MPSSDEGB
कौशल से रोजगार, समृद्धि अपार

FITT
Foundation For Innovation And Technology Transfer

SANKALP
Ministry of Skill Development & Entrepreneurship

Sample Capstone Project Template
Course: AI Builder

Title: Prediction of Renal Diseases using Machine learning

Date:

Course: AI Builder
Course Code:
Mentor Name:

Presented by: List the Names
Name: <Name>
Regn. ID:<Regn. ID>

Overview

Introduction

- The role of kidneys in daily life
  - Responsible for purifying blood and removing toxins from the body
- Reasons for kidney malfunction or failure
  - Kidney tumours (renal tumours), which can be benign or malignant (renal cell carcinoma)
  - Renal cysts, fluid collections in or around the kidney, some of which can be cancerous or linked to polycystic kidney disease
  - Nephrolithiasis (kidney stone disease) caused by calcium deposits in the kidneys or urinary tract
- Symptoms and complications of kidney diseases
  - Abdominal pain and hematuria (blood in urine) may be indicative of kidney tumours
  - Renal cysts and kidney stones can cause abdominal pain and difficulty in passing urine
- Challenges in diagnosing kidney diseases
  - Early stages of kidney diseases may not be clearly visible in CT scans, leading to missed diagnoses
  - Physicians rely on patient symptoms and CT scans, but additional tests like urinalysis and blood culture may be needed
  - Time-consuming process with potential for prolonged patient suffering
- Importance of early diagnosis for renal cell carcinoma
  - Rapid diagnosis crucial to prevent metastasis and improve treatment outcomes

Contribution by team members

- Write Member Name and Work done

Project Plan

Project Schedule

Table 1. Project Schedule

Sl. No.	Phase	Tasks	Start on Day	Duration (Days)
a	Literature Survey	Identify relevant research	1	5
b		Read and analyse the papers	6	5
c		Summary of literature survey	11	5
d	Data collection and preprocessing	Data Collection	16	3
e		Data labelling, cleaning, augmentation	19	3
f	Model training and evaluation (5- and 10-fold CV)	VGG16 (Pre-trained)	22	5
g		MobileNet	27	5
h		ResNet50 (Pre-trained)	32	5
i		InceptionV3 (Pre-trained)	37	5
j		SqueezeNet	42	5
k		Analysis of results	47	5

Project Schedule (Contd.)

l	Novel architecture development	Design and implement the novel architecture	52	7
m		Hyperparameter tuning	59	7
n	Documentation	Draft report preparation	66	10
o		Final report preparation	76	12
		Total Duration	88 days	

Gantt Chart

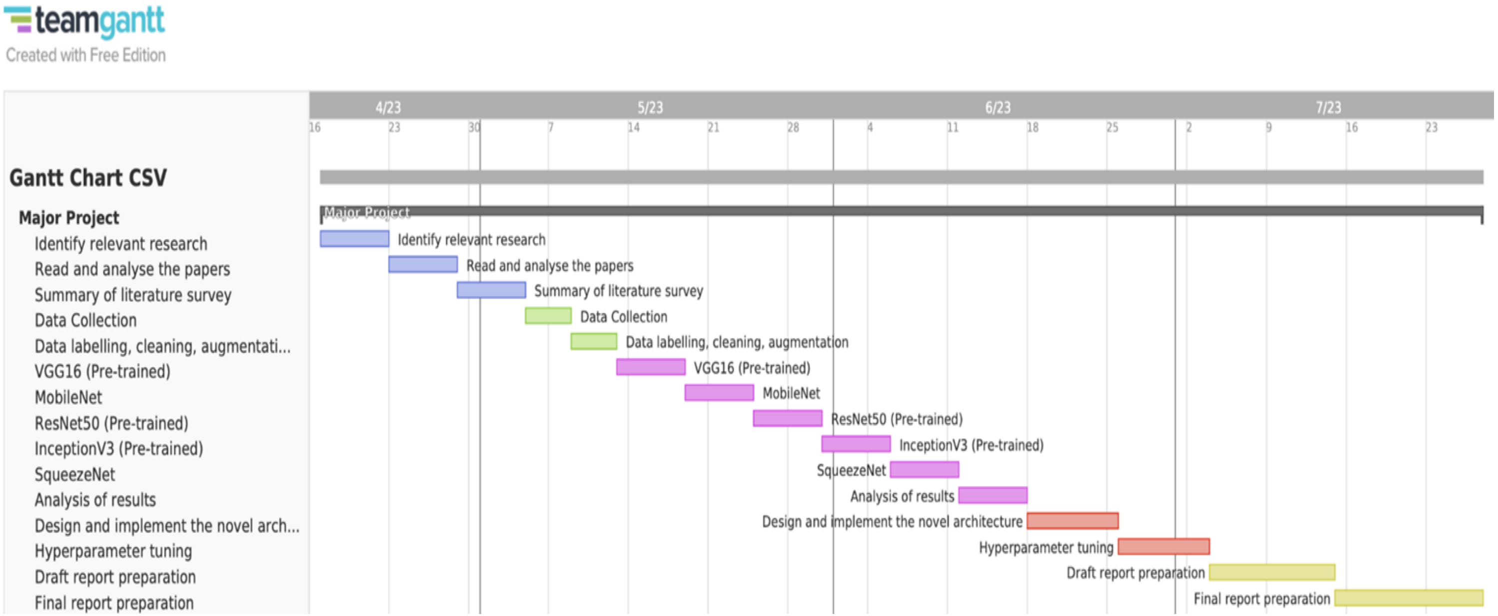


Fig 1. Gantt Chart

Objectives

- To Developing a deep learning architecture to classify whole abdominal CT images into various categories such as
  - Normal
  - Cysted
  - Tumourous
  - Calcified (Stone)
- Comparing the performance of the developed architecture with other legacy deep learning classifiers.
- Review the results for various hyperparameters.

Methodology

Data Collection

- Data publicly available was collected
- Dataset Details
  - Large image dataset with 12,446 CT scan images
  - Images are in DICOM format converted to lossless JPEG
  - Class distribution
    - Cysts 3,709 images
    - Kidney stones (various sizes and shapes) 1,377 images
    - Kidney tumours 2,283 images
    - Structurally-normal kidneys 5,077 images
  - Images include coronal and axial cuts
  - Data Source
    - Data collected from PACS (Picture Archiving and Communication System)
    - Patients were already diagnosed with kidney tumours, cysts, normal conditions, or kidney stones
- Only axial images were selected

Data Preprocessing and Augmentation

- Image Preprocessing
  - Images renamed with class labels as prefixes
  - Resolution reduced to 512x512, maintaining data integrity
  - Even class distribution ensured with 1,000 images per class randomly picked
  - Stone class had only 848 images; image augmentation used to create 152 new images
  - All images resized to 224x224 resolution
  - Filters applied to enhance edges and increase contrast for better model identification of margins
- Dataset Formation
  - Merged augmented images with 3,000 randomly picked images, creating a dataset of 4,000 images (224x224)
- Data Normalisation
  - Images converted to NumPy arrays and stored as binary files
  - MinMax Scalar normalisation applied, reducing pixel values to the range of [-1, 1]
  - Reduced memory usage for deep learning model implementation
  - NumPy array shape (4000, 224, 224, 3) for images, (4000,) for class labels
- External Validation Data
  - 200 images randomly picked for each class, forming an external validation set
  - Images normalised within the range of [-1, 1] and converted to NumPy arrays
  - Resultant arrays shape (800, 224, 224, 3) for images, (800,) for class labels
- Model Training and Evaluation
  - Binary files used directly during model training and evaluation

Sample Images

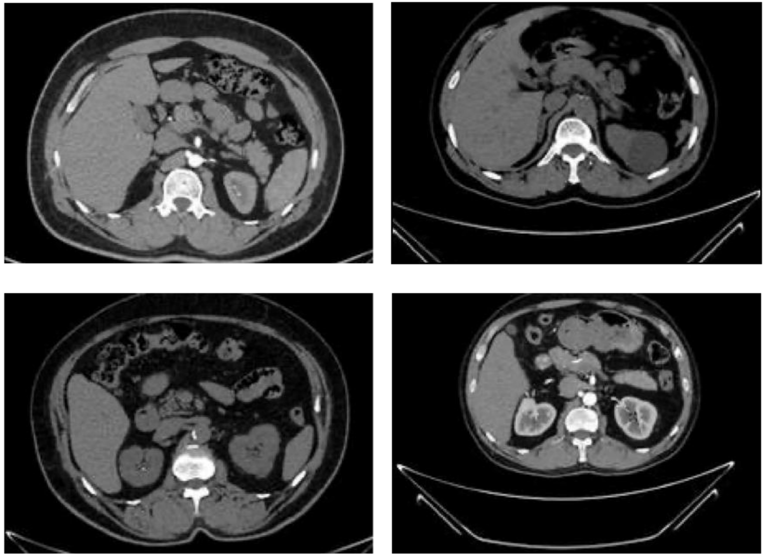


Fig 3. Sample Images (Clockwise from top left: Normal, Cyst, Stone, and Tumour

Model Training

- Training Process
  - Multiple models trained on the preprocessed dataset
  - Two different training strategies used 5-fold and 10-fold stratified cross-validation
- 5-fold Stratified Cross-Validation
  - For each iteration, models trained on four folds of training data
  - Remaining fold split equally into testing and validation sets
- 10-fold Stratified Cross-Validation
  - Dataset split into ten equal 'folds’
  - Each iteration, models trained on nine folds, one fold split into validation and test sets
- Training Details
  - All models trained for 25 epochs per fold of training
  - Random seed value set to 1 for result reproducibility
  - Stratification ensured equal class distribution

Model Training (Contd.)

- Data Preparation
  - Label data converted into a 4x4 matrix using onehot encoding
  - Onehot encoding represents categorical variables as numerical values
- Model Evaluation
  - Stratified k-fold cross-validation ensured robust training data
  - Weights with highest validation accuracy stored in .h5 file during training
- Optimizers and Learning Rates
  - Different optimizers and learning rates used for different models
  - All models trained with a batch size of 64
- Loss Function
  - Models compiled using categorical cross-entropy as the loss function

Model Evaluation

- Model Performance Evaluation
  - Accuracy
  - Precision
  - Recall
  - F1-Score
  - Area under ROC Curve
- Metrics evaluated during testing process in each fold

Review of the Models used

MobileNet

- MobileNet Overview
  - Lightweight and efficient deep learning model optimized for mobile and resource-constrained devices
  - Developed by Google to tackle challenges of limited computational power and memory
- Depthwise Separable Convolution (DSC)
  - Key innovation in MobileNet
  - Splits convolution into depthwise and pointwise steps
  - Depthwise convolution applies a single filter per input channel, reducing computational complexity
  - Pointwise convolution (1x1) linearly combines depthwise outputs to increase output channels
  - Efficiently captures spatial and channel-wise information
- Hyperparameters Width Multiplier (α) and Resolution Multiplier (ρ)
  - α scales input and output channels for device-specific model adjustments
  - ρ scales down input image spatial resolution, reducing computational power requirements
  - Enables MobileNet optimization for different devices and applications

MobileNet (Contd.)

- Fully Convolutional Architecture
  - No traditional fully-connected layers; uses global average pooling instead
  - Global average pooling feeds directly into a softmax activation layer for classification
  - Enables the model to be suitable for transfer learning
- Modified MobileNet
  - Includes three fully-connected layers with dropout and L1 regularization
  - First fully-connected layer has 256 neurons with L1 regularization to prevent overfitting
  - Dropout layer with 50% dropout rate to improve generalization
  - Second fully-connected layer with 16 neurons using ReLU activation
  - Final classification layer with 4 neurons using softmax activation
- Model Parameters
  - Total parameters 3,495,444
  - Trainable parameters 3,473,556
  - No pre-trained weights used
- Model Compilation
  - Compiled using Adam optimizer with initial learning rate of 0.0001

ResNet50

- Overview
  - Belongs to the family of residual networks
  - Introduced by Microsoft Research in 2015 for computer vision tasks
  - Total of 50 layers, with 48 two-dimensional convolution layers with residual connections
  - Uses softmax activation for multi-class classification
- Unique Design
  - Learning of residual functions to address vanishing gradient problem
  - Captures underlying features effectively, leading to superior performance
  - User re-parameterization allows independent learning of each residual block
  - Pre-activation Residual Units enhance gradient flow during training

ResNet50 (Contd.)

- Pre-trained Weights
  - Convolutional layers use pre-trained weights
  - Only the classification layer is trained for specific problem
- Fully-Connected Layer
  - 4 neurons with softmax activation after global average pooling
- Optimization
  - Stochastic Gradient Descent (SGD) used as the optimizer
  - Initial learning rate set to 0.001
- Model Parameters
  - Total parameters 23,595,908
  - Trainable parameters 8,196

Proposed Architecture (Optional)

- Overview
  - Total of 19 layers
  - Input tensor passed through 2D convolutional layer with 64 filters, 7x7 kernel, stride (2, 2), and ReLU activation
  - MaxPooling2D layer with 3x3 kernel and stride (2, 2) for downsampling
- Residual Blocks
  - Three blocks with two 2D convolutional layers (3x3 kernel) and ReLU activation
  - Shortcut connections to mitigate vanishing gradient problem
- Depthwise Separable Convolution
  - Applied with 128 filters, 3x3 kernel, and stride 1
  - Decomposes standard convolution into depthwise and pointwise convolutions
- Fire Modules
  - Two modules (16, 64) and (32, 128)
  - Each module 1x1 convolution (squeeze) followed by two branches (1x1 and 3x3 convolution)
  - Outputs concatenated for next layer input

Proposed Architecture (Contd.)

- 2D Convolutional Layer
  - Applied with 256 filters, 3x3 kernel, and ReLU activation
  - MaxPooling2D layer (2x2 kernel, stride (2, 2)) for further downsampling
- GlobalAveragePooling2D
  - Used to pool spatial information and produce 1D vector
- Dense Layer
  - 4 neurons for final classification output (4 output classes) with softmax activation
- Performance Benefits
  - Utilizes residual connections, depthwise separable convolutions, and fire modules
  - Achieves high performance with efficient computational resource usage
  - Suitable for image classification tasks on resource-constrained devices
- Model Compilation
  - Compiled using Adam optimizer with an initial learning rate of 0.001
- Model Parameters
  - Total parameters 889,268
  - Trainable parameters 889,268
  - Non-trainable parameters 384

Experimental Results

Model Description

Table 3. Model Description

Sl. No.	Model	Transfer Learning?	Optimiser	Learning Rate	Number of parameters	Number of trainable parameters
1	VGG16	Yes	SGD	0.001	14716740	2052
2	MobileNet	No	SGD	0.0001	3495444	3473556
3	ResNet50	Yes	SGD	0.001	23595908	8196
4	InceptionV3	Yes	Adam	0.001	21810980	8196
5	SqueezeNet	No	Adam	NA	726636	726636
6	Proposed Architecture	No	Adam	0.001	889652	889268

Model Description (Contd.)

Table 4. Training Time

Sl. No.	Model	Training Time (Mins) (5-fold CV)	Training Time (Mins) (10-fold CV)
1	VGG16	50.94	72.4
2	MobileNet	35.69	45.3
3	ResNet50	39.09	50.2
4	InceptionV3	26.3	37.2
5	SqueezeNet	18.06	30
6	Proposed Architecture	37	49

5-fold Cross-Validation Results

Table 5. 5-fold CV Results

Sl. No.	Model	Average Accuracy	Average Precision	Average Recall	Average F1-Score	Average AUROC
1	VGG16	0.9198	0.9228	0.9206	0.9192	0.947
2	MobileNet	0.9964	0.9966	0.9966	0.9966	0.9978
3	ResNet50	0.7248	0.7334	0.7248	0.7234	0.8162
4	InceptionV3	0.9972	0.9968	0.997	0.9968	0.9978
5	SqueezeNet	0.2514	0.0628	0.25	0.1004	0.5
6	**Proposed Architecture**	**0.9992**	**0.999**	**0.999**	**0.999**	**0.9992**

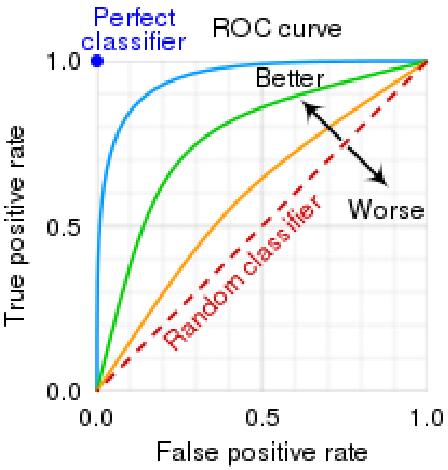
10-fold Cross-Validation Results

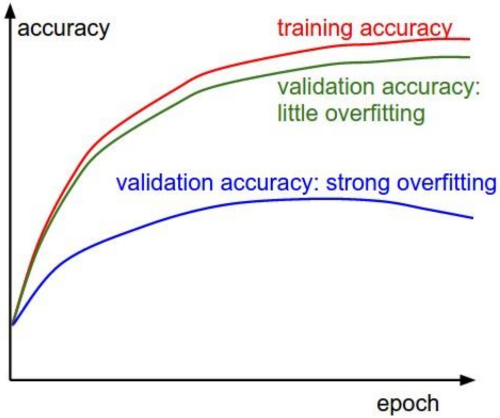
Table 6. 10-fold CV Results

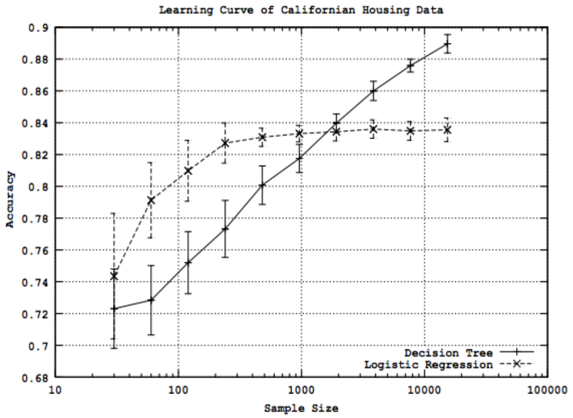
Sl. No.	Model	Average Accuracy	Average Precision	Average Recall	Average F1-Score	Average AUROC
1	VGG16	0.9	0.9018	0.9006	0.8989	0.9338
2	MobileNet	0.995	0.995	0.9952	0.995	0.9967
3	ResNet50	0.7495	0.7565	0.7494	0.7479	0.8329
4	InceptionV3	0.9965	0.9963	0.9965	0.9964	0.9978
5	SqueezeNet	0.225	0.0566	0.25	0.0914	0.5
6	**Proposed Architecture**	**0.997**	**0.9923**	**0.9867**	**0.99**	**0.998**

ROC curves, Learning Curves

➢ Use charts, line graphs, Learning curves, ROC curves, and interpret them.







Conclusion

- Compared and contrasted deep learning models for renal disease prediction and built novel architectures for improved accuracy
- Literature survey laid the foundation for model selection with state-of-the-art techniques
- 4800 CT scan images split into training (4000) and external validation sets (800)
- Utilized 5 and 10-fold cross-validation for rigorous training and evaluation
- Models demonstrated 4-class classification capabilities (Normal, Cyst, Stone, Tumour)
- A new architecture having 3 residual blocks, a DSC block, 2 fire modules, a convolution layer with max-pooling, and dense layer was proposed
- Accurate renal disease prediction improves patient outcomes and healthcare efficiency
- Early detection and precise classification lead to timely interventions and better treatment planning
- Potential benchmarks for future research and innovations in medical image analysis and deep learning
- Lasting positive impact on healthcare practices and inspiration for future medical deep learning research

References

- Patro, K. K., Prakash, A. J., Neelapu, B. C., Tadeusiewicz, R., Acharya, U. R., Hammad, M., Yildirim, O., & Pławiak, P. (2023). Application of Kronecker convolutions in deep learning technique for automated detection of kidney stones with coronal CT images. Information Sciences, 119005. https://doi.org/10.1016/j.ins.2023.119005
- Liu, Y.-Y.; Huang, Z.-H.; Huang, K.-W. Deep Learning Model for Computer-Aided Diagnosis of Urolithiasis Detection from Kidney–Ureter–Bladder Images. Bioengineering 2022, 9, 811. https://doi.org/10.3390/bioengineering9120811
- Chaitanya, S.M.K., Rajesh Kumar, P. (2019). Detection of Chronic Kidney Disease by Using Artificial Neural Networks and Gravitational Search Algorithm. In: Saini, H., Singh, R., Patel, V., Santhi, K., Ranganayakulu, S. (eds) Innovations in Electronics and Communication Engineering. Lecture Notes in Networks and Systems, vol 33. Springer, Singapore. https://doi.org/10.1007/978-981-10-8204-7_44
- Yang, E., Kim, C. K., Guan, Y., Koo, B., & Kim, J. (2022). 3D multi-scale residual fully convolutional neural network for segmentation of extremely large-sized kidney tumor. Computer Methods and Programs in Biomedicine, 215, 106616. https://doi.org/10.1016/j.cmpb.2022.106616
- Zhang, H., Botler, M., & Kooman, J. P. (2023). Deep Learning for Image Analysis in Kidney Care. Advances in Kidney Disease and Health, 30(1), 25-32. https://doi.org/10.1053/j.akdh.2022.11.003
- Uhm, KH., Jung, SW., Choi, M.H. et al. Deep learning for end-to-end kidney cancer diagnosis on multi-phase abdominal computed tomography. npj Precis. Onc. 5, 54 (2021). https://doi.org/10.1038/s41698-021-00195-y
- George, M., Anita, H.B. (2022). Analysis of Kidney Ultrasound Images Using Deep Learning and Machine Learning Techniques: A Review. In: Ranganathan, G., Bestak, R., Palanisamy, R., Rocha, Á. (eds) Pervasive Computing and Social Networking. Lecture Notes in Networks and Systems, vol 317. Springer, Singapore. https://doi.org/10.1007/978-981-16-5640-8_15