**Deep Learning Project Chronic Kidney Classification**

Project flow

1. Project templet creation
2. Project setup and requirement installation
3. Logging, Utils and exception Module
4. Project workflow
5. Notebook Experiment
6. Modular code implementation
7. Training pipeline
8. MLflow (MlOPs tool) – For experiment tracking and model registration
9. DVC (MLOps tool) – for pipeline tracking and implementation
   1. DVC : Data version control
10. Prediction pipeline and User app creation
11. Docker
12. Final CI/CD Development

Tack:

Build deep learning base Image classification model which able to classify, CT scan image of kidney and identify kidney with tumour and Normal kidney.

Dataset:

I am using dataset of Kidney CT Scan image, which is identified and classified as kidney with tumour and Normal kidney with respective to image.

For example:



Normal kidney

A close-up of an x-ray of a person's body

Description automatically generated

Tumour Kidney

Download Database:

<https://drive.google.com/file/d/1BIiJfOdei5xm9XMw_Gq9Bdq8uQWFtSPC/view?usp=sharing>

Or

Full volume Dataset: [CT KIDNEY DATASET: Normal-Cyst-Tumor and Stone (kaggle.com)](https://www.kaggle.com/datasets/nazmul0087/ct-kidney-dataset-normal-cyst-tumor-and-stone)

step 1.

Templet.py file created, which help to develop required file and folder for project.

Installing git in project env.: *conda install -c anaconda git*

Package requirement:

tensorflow==2.12.0

pandas

dvc

mlflow==2.2.2

notebook

numpy

matplotlib

seaborn

python-box==6.0.2

pyYAML

tqdm

ensure==1.0.2

joblib

types-PyYAML

scipy

Flask

Flask-Cors

gdown

-e .

3) Added logging functionally, it help to detect bug in code and we can observer how code functioning

4) Modular codding approach

5) setup DVC pipeline

**Future improvement**

**- Enable gpu training**

**- Re-Define project all research adds in single notebook and then build module**

**- Train model on full data with all class.**

**- Automate process of copy model.h5 from training dir to model dir for GitHub upload using python**