INTERNSHIP REPORT

A report submitted in partial fulfilment of the requirements for the Award of Degree of

In

Post Graduate Program in Data Analytics and Data Science

by

ALPHIN GNANARAJ I

Under Supervision of

MR. SAURABH RANJAN

(Duration: 25th December, 2022 to 25th January, 2023)

CATARACT DETECTION USING CNN & IMAGE CLASSIFICATION

Project Insight:

1. Introduction.
2. Packages & ML models.
3. Code.
4. Github platform.
5. Steps to view output.
6. Conclusion.

1. Introduction:

Cataract is a clouding of the natural lens in the eye that affects vision. Early detection and treatment of cataracts are important for maintaining good vision and quality of life.

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that have been successfully applied in various fields, including computer vision. In the context of cataract detection, a CNN can be trained to analyse eye images and identify the presence of a cataract.

2. Packages & ML models:

- **NumPy:** It is a core scientific computing library, provides efficient multi-dimensional array objects and various operations to work with these array objects. It's used to perform a wide variety of mathematical operations on arrays.

- **Pandas:** Library for data analysis provides high-performance data manipulation in Python, work with data frame. It's used for data cleaning, normalization, visualization, inspection etc.

- **Random:** A built-in module that provides a suite of functions for working with random numbers. These functions allow you to **generate random numbers, select random elements from a list, shuffle elements in a list and perform other random operations**. Some of the most commonly used functions in the random module are:

    - random.random(): generates a random float number between 0 and 1 (both inclusive of 0, but not 1).
    - random.randint(a, b): generates a random integer between a and b (both inclusive).
    - random.randrange(start, stop, step): generates a random number in the given range, with an optional step.
    - random.choices(iterable, weights=None, *, cum_weights=None, k=1): returns a list with k randomly selected elements from the iterable, with optional weights for each element.

- random.shuffle(iterable): shuffles elements in a list in place and returns None.
- random.sample(population, k): returns a list of unique elements randomly selected from the population, without replacement.

- **Tensorflow:**
  - It is used for a wide range of tasks such as building and training neural networks, natural language processing, image and video processing, and more.

  - TensorFlow provides a high-level API for building and deploying machine learning models, as well as a low-level API for defining and manipulating tensors (multidimensional arrays) and mathematical operations.

- **Tensorflow.keras.preprocessing.image:**
  - The ImageDataGenerator class in TensorFlow's keras.preprocessing.image module is used for data augmentation in computer vision tasks, such as image classification and object detection.
  - Data augmentation involves generating additional training examples by applying various transformations to the existing training images, such as rotation, scaling, flipping, and cropping. This is useful to prevent overfitting, increase the size of the training data, and improve the generalization ability of the model.
  - ImageDataGenerator in TensorFlow provides a convenient way to perform data augmentation on image data, by specifying the desired transformations using arguments.
  - It takes in the training data in the form of a directory containing the image files, and generates augmented data on the fly during training. The generator yields batches of augmented images and labels that can be fed into a deep learning model for training.

- **CV2:** The cv2 module in Python is a wrapper for OpenCV (Open Source Computer Vision Library). OpenCV is a popular computer vision library that provides a vast collection of algorithms and functions for image and video processing. The cv2 module makes it easy to use OpenCV in Python by providing a Pythonic interface to OpenCV's C++ code. The cv2 module provides a wide range of functions for image and video processing, including:
  - Image loading and saving.
  - Image filtering and transformation, such as color space conversions, thresholding, and morphological operations.
  - Feature detection and description, including SIFT, SURF, and ORB.
  - Object detection and tracking, including Haar cascades and HOG-based object detection.
  - Camera calibration and 3D reconstruction.

- **Matplotlib: Matplotlib.pyplot** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an interface for creating plots, histograms, bar charts, error charts, scatterplots, etc. **pyplot** provides a high-level interface for creating plots and manipulating visualizations, making it easy to generate publication-quality visualizations.

- **Tqdm:** Tqdm is a library that provides progress bars for loops and iterables in the command line. It is useful for tracking the progress of long-running tasks, such as training a deep learning model or loading a large dataset. The library works by wrapping an iterable object and displaying a progress bar that updates in real-time as the iteration progresses. This helps to give an estimate of how long a task will take to complete and provides feedback on progress.
- **OS:** The ‘os’ module in Python provides a way to interact with the underlying operating system. It provides functions for reading and writing files, accessing and manipulating the file system, starting new processes, and accessing environment variables, among other things. The os module is a built-in Python library and is commonly used in Python scripts and programs to perform a wide range of system-related tasks.

- **Tkinter:** Standard GUI library for Python. It provides a way to create graphical user interfaces (GUIs) for desktop applications. tkinter is easy to use and provides a range of widgets, such as buttons, labels, and text boxes, that can be combined to create complex applications.

- **Keras.model**: Keras is a high-level deep learning library for Python that provides a simple interface for building and training neural networks. keras.models is a submodule of the keras library that provides a way to create, compile, and train deep learning models. It provides a high-level API for building complex models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), with minimal code.

- **PIL (Python Imaging Library):** PIL is a library that provides a way to work with images in Python. It provides a range of functions for opening and saving images, cropping, resizing, and transforming images, among other things. PIL is commonly used in computer vision projects for loading and processing image data, and it is also used in image processing tasks such as image resizing, image cropping, and color correction.

3. Code:

As the code is too large, I'm attaching the link:
https://github.com/Alphin62/CNN---Cataract-Detection-using-GUI

4. Github platform:

- **Github:** GitHub is a web-based platform that provides hosting for software development and version control using Git. It allows developers to store, share, and collaborate on code, as well as track and manage changes to the code.

Key features of GitHub:

- **Version control:** GitHub allows developers to save different versions of their code, known as "commits", and track changes over time. This makes it easy for developers to roll back to previous versions of the code if needed.

- **Issues and pull requests:** GitHub provides a built-in system for tracking and managing bugs and feature requests, known as "issues". Developers can also submit "pull requests" to propose changes to the code.

- **Open-source:** A large number of projects on GitHub are open-source, meaning that the source code is available for anyone to view, use, and contribute to.

5. Steps to view Output:

- Keep the required files in a folder, run both ".ipynb" files.
- After executing 'GUI code.ipynb', a screen will appear.
- Click on Browse, select the fundus image.
- The output will be displayed whether Cataract or not.

6. Conclusion:

- A deep learning approach using Convolutional Neural Networks (CNNs) can effectively detect cataracts in eye images. CNNs have been shown to outperform traditional image processing techniques in terms of accuracy and efficiency.
- The use of CNNs in cataract detection can assist ophthalmologists in making a more accurate diagnosis and improve patient outcomes.
- However, the results of a specific project may vary based on the quality and quantity of data used, the design of the network, and the choice of evaluation metrics. Further research and refinement of CNNs for cataract detection can lead to more reliable and practical applications in clinical settings.