

Take-Home Assignment: Age Prediction from Facial Images

Objective: Develop a deep learning model to predict age from facial images.

Dataset:

The dataset for this project can be found [here](#). It consists of images of people between 20 and 50 years old, categorised by age in separate folders. Each folder name represents the age of the individuals in the images.

Instructions:

1. **Data Analysis:**
 - Download the dataset.
 - Explore the dataset to understand image distribution, statistics, and visualise samples for insights.
2. **Data Cleaning:**
 - Implement a method to remove low-quality images from the dataset. Define your criteria for "low quality" (e.g., blurry images, ...).
 - Document the implemented cleaning process and the number of images removed.
3. **Model Development:**
 - Train a deep learning model to predict age from the images.
 - **Important:** Since the model needs to be deployed on mobile devices, make sure that it has a maximum of 1 million parameters.
4. **Training and Evaluation:**
 - Train the model for a minimum of 20 epochs, model performance is not the focus of this assignment.
 - Evaluate and report model performance.
 - Discuss the chosen evaluation metric/s and its/their significance in this context.
5. **Model Packaging for C++ Deployment:**
 - Imagine you are handing over the model to a C++ developer. Package the trained model and any necessary scripts (in Python) for seamless integration into a C++ pipeline.
 - Briefly explain the steps involved in using your packaged model for age prediction in C++.
 - **Note:** You don't need to write actual C++ code, just explain the process conceptually.
6. **Future Work:**
 - Discuss potential improvements to the model or the entire pipeline.

Evaluation Criteria:

- **Code Readability:** Your code should be well-structured, documented, and easy to understand for someone unfamiliar with your project.
- **Thought Process:** Document your thought process for each step.
- **Documentation:** Provide clear and concise documentation for all code sections, data cleaning procedures, and model packaging steps.

Additional Notes:

- Feel free to use any libraries or frameworks suitable for image processing, model development, and deployment.
- Include comments in your code to explain functionalities and decisions made.
- Submit your code, documentation, and report using a cloud-based version control of choice like GitHub or GitLab.

This take-home test focuses more on understanding your thought process and approach to building the pipeline rather than achieving top model performance.