

Assignment No. 3

1 TITLE

Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the problem statement to identify objects, morphisms, overloading in functions (if any) and functional relations and any other dependencies.

2 AIM

To identify object oriented concepts for our project.

3 OBJECTIVE

To exploit distributed/parallel/concurrent processing of the problem statement To identify objects, morphisms, overloading in functions (if any).

4 THEORY

1. Divide and conquer strategies to exploit distributed/parallel/concurrent processing of the problem statement : In computer science, divide and conquer is an algorithm based on multi-branched recursion. We use divide and conquer strategies such as: We have divided the problem into classes and functions, and then further we merged them together to generate result. Here, we are making pre-processing, low level feature extraction, high level feature extraction, and then we finally connect all these modules to achieve the desired goal of our problem.
2. Objects : Objects are entities having some identifiable attributes and methods that represent behaviour of real world entities. Objects are instances of class. In this project, we consider the following features as objects :

- Face coordinates
 - Facial features matrix
 - Voice segments
 - Voice feature matrix
 - Speech text
 - Response
3. Functional dependencies : A function 'A' is said to be functionally dependent on function 'B' if A's input is the output of 'B'.
 - Input module capturing video frames and voice segments
 - Communication module which relays input processed output between the robot and server
 - Facial Recognition
 - Voice Recognition
 - Speech to Text
 - Emotion analysis from facial features
 - Emotion analysis from voice
 - Emotion analysis from speech text
 - Retrieval based response system

All these modules are dependent on each other and need to be followed in proper order.

5 PARALLELISM IN THE PROJECT

1. A heirarchy of flow of control among the modules is show in the figure-1 .
2. Modules at the same level in the heirarchy are functionally independent.
3. Data dependencies, if they emerge among the modules can be handled by mutually exclusive access or by increasing redundancy of data, making the replicated data private to each thread.
4. Mututally exclusive access can be enforced with thread safe functions or by implementing monitors.

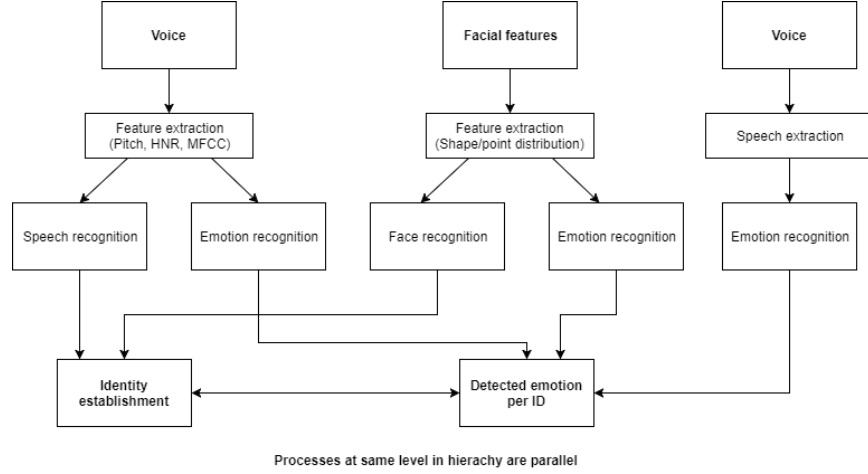


Figure 1: Parallel modules in the project

5. Increasing redundancy of data can result in inconsistent states, if the data is mutable. Redundancy is to be implemented only for immutable data, that is only to be read, such as audio and video frames.
6. Modules at the next level of hierarchy are blocked until all the modules at the current level have finished execution.
7. Thus modules are functionally dependent at consecutive levels.
8. Feature extraction is common to identity recognition and emotion recognition tasks. For voice, handcrafted features which are extracted such as pitch characteristics, Harmonics to Noise Ratio, Mel-Frequency Cepstral Coefficients are required for both identity recognition and emotion recognition using speech. Similar for facial features, extracted features are common for identity and emotional recognition tasks.
9. Identity and emotion recognition tasks in each module can be performed parallelly across the modules as they are functionally independent.
10. Establishment of identity using outputs from all modules and linking recognised emotion state is performed at the next level. This level cannot be processed until all the tasks in the previous level have concluded. Thus level-to-level flow has functional dependency.