Face Recognition based Attendance System

Problem Statement

Most people will agree with the fact that attendance is one of the most important activities in the class and thus should be dealt with very seriously.

At IIIT we have tried several ways to overcome this challenge. In the early days, we worked out with teachers calling out names at the beginning or end of the class. We then realized that all of this load can be shifted to a single designated person, thus for a long time, we had Badri who although is a hero to many but let's face it, a lot of times breaks the class flow, which can become a hurdle in learning. On top of that, it is followed by making manual digital entries of the same. Clearly this process is tedious and has a lot of scope in automation.

Then we had a biometric system, slightly more advanced but still very distracting and time-consuming.

IIIT has one of the best computer vision and ML labs of the country and hence we must come up with a better system to automate all of this and save a lot of resources.

In today's digital AI-powered world, all of this can be automated by the virtue of Deep Learning.

We have come up with a completely automated system that uses facial recognition to mark attendance. This will save a lot of man-hours and will be a lot more efficient. There has been a lot of research in this domain and there exist some very efficient and accurate algorithms that are tailor-made for face recognition. We will be making use of such algorithms to extract face and match it with the existing student images on Moodle. The face when matches with someone, he/she is marked present directly on the attendance server.

Another advantage is that since this a very active research area, we can continuously upgrade our models for a fool-proof and efficient system using state of the art techniques of computer vision and deep learning.

The hardware requirements are minimal too, as we just need a webcam and all the processing will happen on a remote server.

Formally, we aim to develop an Automatic Face Recognition based Attendance System wherein a student will be marked present/absent based on whether or not his/her face was detected.

Technology used:

- One of the aspects that makes python such a popular choice in general, is its abundance of libraries and frameworks that facilitate coding and save development time.

- Python is an open-source programming language and is supported by a lot of resources and high-quality documentation. It also boasts a large and active community of developers willing to provide advice and assistance through all stages of the development process.

- Python is a platform-independent language therefore different nodes of different software stack can work without any problem.

- It is ideal for rapid prototyping.

- All major deep learning frameworks support Python. Of these, the most popular and powerful platforms are TensorFlow, Keras (which is typically used as a front-end wrapper for TensorFlow), and PyTorch.

  - TensorFlow is Google's open-source deep learning framework. It provides powerful capabilities for structuring datasets, working with Tensors (multidimensional arrays, a basic building block of neural networks) and constructing deep learning architectures.

  - Keras is an open-source deep-learning library. It enables fast experimentation by giving developers access to standard neural network models with a simple programming model.

  - PyTorch is a middle ground between TensorFlow and Keras – it is powerful and allows you to manipulate tensors and lower-level constructs, but is also easy to use and provides convenient abstractions that save time.

Solution Overview:

(the diagram is included in the repo)
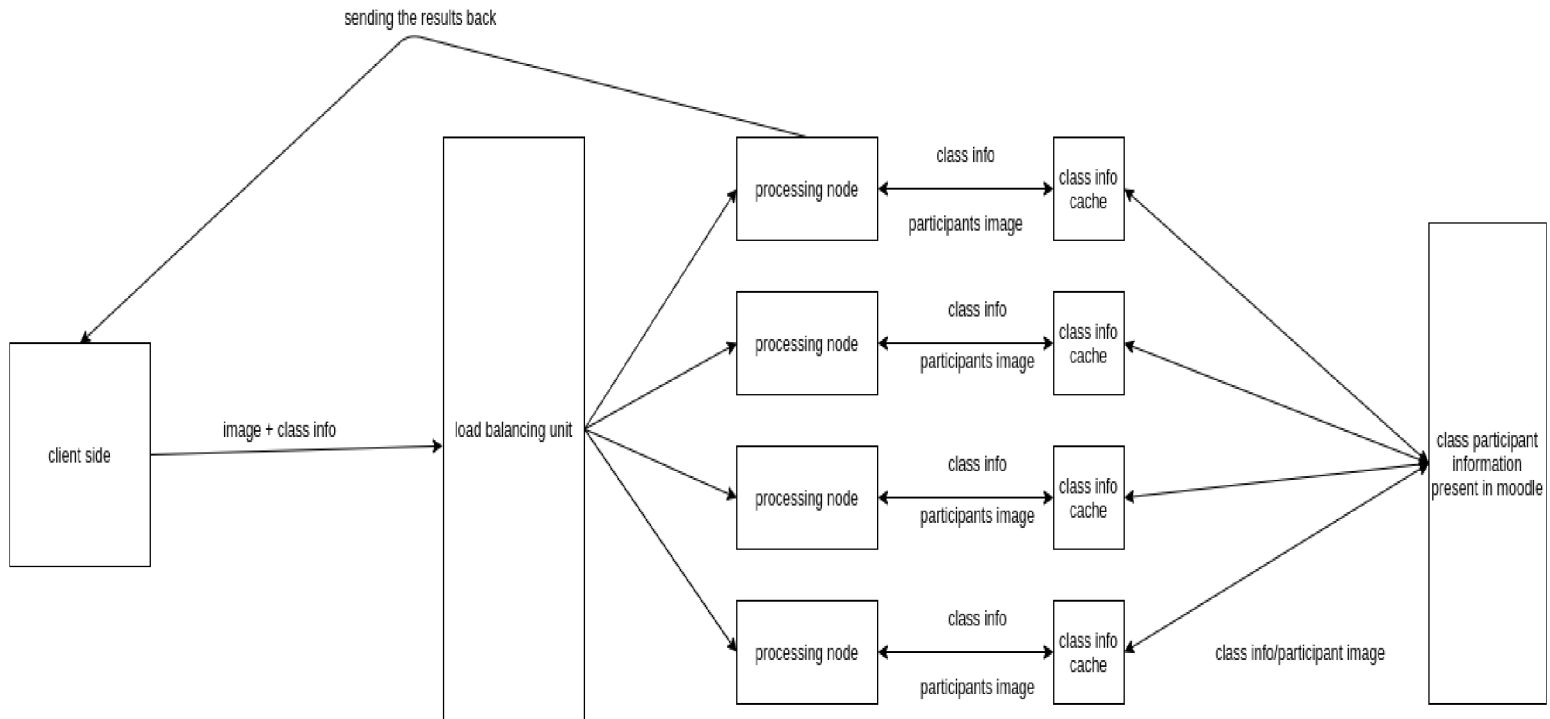
As shown in the above figure the solution will work as follows.

- First, the client-side will send the image and class id of the class to the back end this can be done in various ways for example through a client-side mobile application or through an email. Both methods have their advantages and disadvantages.
- In the back end, the load balancing unit will send the request to the appropriate processing node. This makes the application more scalable and available.
- Now the processing node will detect the faces and match it to the images of the students present in the moodle.
- After doing all the processing the processing node will create an attendance spreadsheet of the class and send it back to the client-side (email or mobile application). All processing nodes can individually send the result back to the respected client i.e no centralized unit is given the responsibility of result delivery this makes the application more scalable and fail-safe.
- Other than the load balancing unit every unit in the solution has replication making the solution robust and scalable.

This design can be modified a bit to add more features as the project progresses.