University Project | Review 2

Attendance System using Deep Learning Face Identification Algorithms

A Project by:

Joel Rego (20171CSE0271)
Medha M H (20171CSE0387)
Mahesha R (20171CSE0359)
Akshay Krishna (20171CSE0036)

Project Guide:

Prof. R C Ravindranath

Faculty, Department of Computer Science and Engineering,
Presidency University, Bangalore

Plan of this presentation

1. Reviewing our project goals (1 - 2 min, by Mahesh)
2. Concepts used in our project (1 -2 min, by Akshay Krishna)
3. Architecture and Block Diagram (3 - 4 min, by Medha M H)
4. Implementation, Algorithms (3 - 4 min, by Joel Rego)
5. Display of code and work done so far.
6. Questions from the panel of reviewers.

1. Let's review our project goals and research gap identification from review 1

> Our project involves use of deep neural networks for face identification.

> Face detection, though complex, is trivial, as it involves only finding faces (labelling face objects). Identifying different faces to a high degree of accuracy is a challenge.

> Our base paper, published in IJML, uses a raspberry pi camera to take photographs for the attendence system. In this project, we aim to simplify this program in python so that it consumes just enough memory to be embedded in an existing device (such as CCTV cameras).

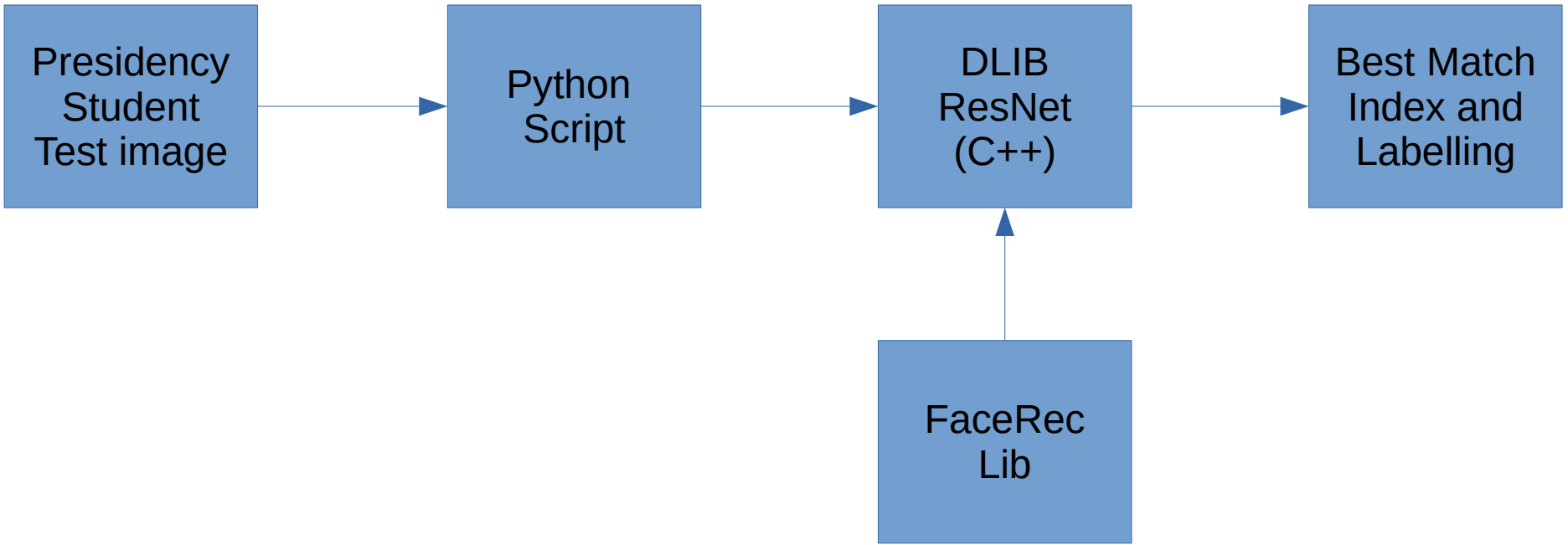
2. Concepts used for our Project

> Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

> In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery.

> ResNet, or Residual Networks is a small twist to Convolutional neural network that has shown ground breaking performance since LSVRC2012 (a global image recognition contest).

3. System Architecture



4. Implementation and Algorithm

4. Algorithm we have used

> Step 1: Encode all images in our database in matrix format.

> Step 2: Create a python dictionary, where the key is equal to name of the person, and value is equal to the matrix.

> Step 3: Encode Test image in matrix format (using same method)

> Step 4: To compare which face matches with the matrices in our dictionary, we will use ResNet (dlib).

> Step 5: To label, we will use the key of the matched index since the key stores the name of the person.