Dynamic Mess Billing System using Face Recognition

Ravi Chopra
2019BCS046

Introduction

1

Context

The objective is to develop and experiment with solutions to automate and dynamise the billing system in the mess areas of the institute. The aim is to use a face recognition module to achieve the above-stated objectives

Problem/Motivation

- The problem with the existing billing system is that it bills all the students the same even if they skip some meals due to some reasons.
- There is no way for our mess workers to predict how much food to prepare, which leads to a lot of food wastage.

Face Recognition Problems

- Most existing face recognition systems suffer from problems like face orientation in the image, varying brightness, noise, etc. Some models also suffer from the problem of overfitting due to the non-linearity of the cosine function.

Objectives

- **Face Recognition module:** The model must be robust to changes in brightness, orientation and other factors.
- **Web Interface:** A web interface for students to interact with the system.
- **Deployment:** Optionally, we are thinking to deploy the face recognition model on Raspberry Pi.

Work Flow

- **Step 1:** Developing/Improving an existing model for face recognition that is not sensitive to changes to brightness and orientation.
- **Step 2:** Developing a web interface for students to interact with system.
- **Step 3:** Using requests module in python to help the model interact with web application’s server side.
- **Step 4:** Using Twilio API to send notification messages to students.
- **Step 5:** Using Payments API to allow students clear their dues.
- **Step 6(Optional):** Deployment on Raspberry Pi.

Literature Review

2

Literature read

- ArcFace: Additive Angular Margin Loss for Deep Face Recognition (9 Feb 2019)
- Mis-classified Vector Guided Softmax Loss for Face Recognition (26 Nov 2019)
- LinCos-Softmax: Learning Angle-Discriminative Face Representations With Linearity-Enhanced Cosine Logits (15 June 2020)

ArcFace Ananlysis

- Add an additive angular margin penalty m between X_i and W _yi to simultaneously enhance the intra-class compactness and inter-class discrepancy. This additive angular margin penalty is equal to the geodesic distance margin penalty in the normalised hypersphere.
- Loss function:

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

MV Softmax Loss Analysis

- This paper explicitly indicates the hard examples as misclassified vectors and adaptively emphasizes them to guide the discriminative feature learning.
- This paper defines a binary indicator I_k to adaptively indicate whether a sample (feature) is misclassified by a specific classifier w_k (where k != y) in the current stage:

$$I_k = \begin{cases} 0, & f\left(m, \theta_{w_y, x}\right) - \cos\left(\theta_{w_k, x}\right) \geq 0 \\ 1, & f\left(m, \theta_{w_y, x}\right) - \cos\left(\theta_{w_k, x}\right) < 0 \end{cases}$$

MV Softmax Loss Analysis

- Loss:

$$\mathcal{L}_5 = -\log \frac{e^{s f(m,\theta_{w_y,x})}}{e^{s f(m,\theta_{w_y,x})} + \sum_{k\neq y}^{K} h\left(t,\theta_{w_k,x},I_k\right) e^{s\cos(\theta_{w_k,x})}}$$

LinCos Analysis

- The nonlinearity of the cosine function may lead to insufficient angular optimization between features and corresponding class weights. As a result, the angular discriminability of the features may be compromised, resulting in a reduced generalization ability.
- The main idea is the use of a linear-cosine logit, which is designed by performing Taylor expansion on a linear logit.

LinCos Analysis

- Loss:

$$\theta_j = \arccos\left(\cos\theta_j\right) \approx \hat{\theta}_j$$

$$\text{where } \hat{\theta}_j = \tfrac{\pi}{2} - \textstyle\sum_{n=0}^{K-1} c_n \left(\cos\theta_j\right)^{2n+1}$$

$$\text{and } c_n = \tfrac{(2n)!}{2^{2n}(n!)^2(2n+1)}$$

$$f_j^{linear} = -\theta_j + \tfrac{\pi}{2} = -\arccos\left(\cos\theta_j\right) + \tfrac{\pi}{2}$$

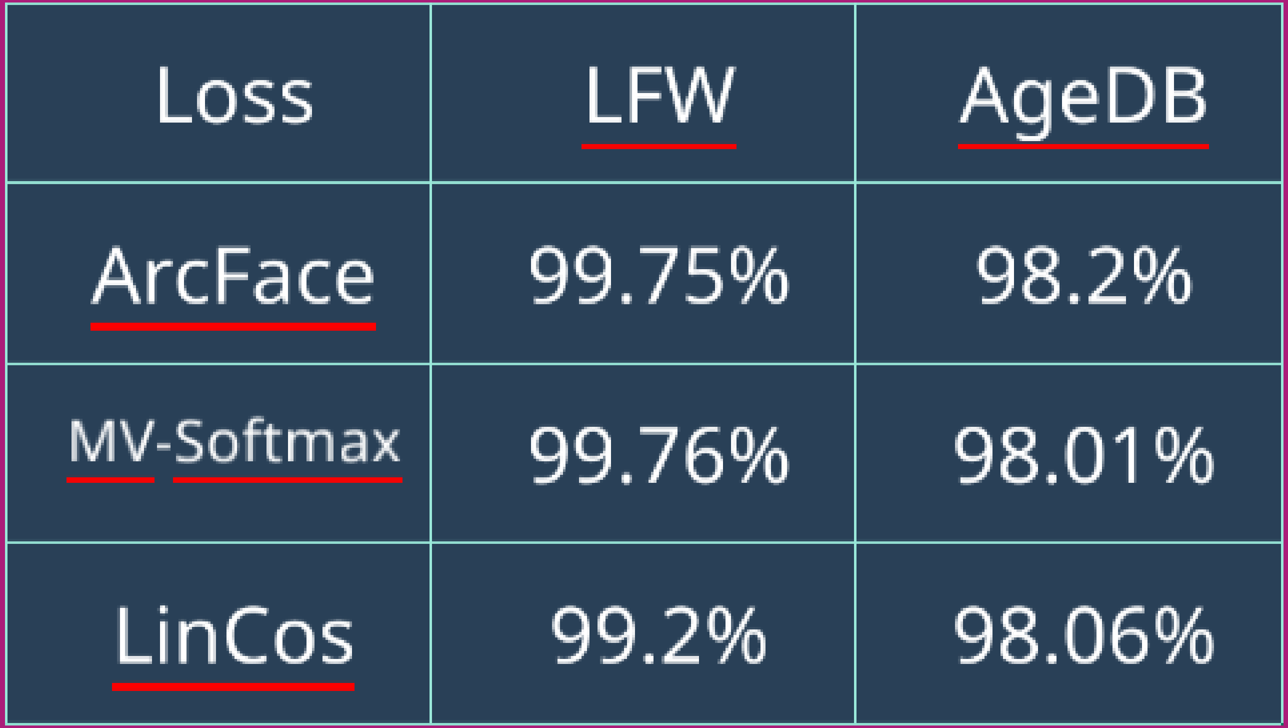
$$f_j^{LinCos} = -\hat{\theta}_j + \tfrac{\pi}{2} = \textstyle\sum_{n=0}^{K-1} c_n \left(\cos\theta_j\right)^{2n+1}$$

$$L_i^{LinCos} = -\log\left(P_{y_i}^{LinCos}\right)$$

$$P_{yi}^{LinCos} = \frac{e^{s f_{yi}\, Lin\ \cos}}{\sum_{j=1}^{C} e^{s f_j^{LinCos}}}$$

Research Gaps

- It is observed that all the algorithms are on their saturation, i.e. it is quite not possible to decide the best of them.



Loss	LFW	AgeDB
ArcFace	99.75%	98.2%
MV-Softmax	99.76%	98.01%
LinCos	99.2%	98.06%

Problem Formulation

- Due to ease of implementation and familiarity with Pytorch, I chose to proceed with ArcFace. Various experiments can be done with the above methods. One of them is to combine the ideas of LinCos and ArcFace. We can approximate cos(θ + m) (where m = Marginal Penalty) using the method proposed in LinCos paper and reduce the problem of over-fitting. Another possible work-around is to replace cosine with the tangent.

Conclusion to literature review

- To conclude, there has been extensive research in the field of face recognition. Due to this, many proposed models have already achieved an accuracy that is possibly less only by the Bayes error. However, there is still room for experimentation, and my focus will remain on trying the above-proposed experiments.

System Requirements

3

System Requirements

- Students are recognised by their faces and mapped to their IDs.
- Students can view their attended meals.
- Students get notified on entering the mess.
- Students can pay and view bills on monthly basis.

Implementation Specifics

- **Pytorch:** It is used as the framework for the development of the face recognition model.
- **MERN stack:** MERN stands for MongoDB database, Express.JS, React.JS, Node.JS. This is the used stack for the web interface.
- **requests:** It is a module in python used for making calls to webserver from face recognition system.
- **Twilio API:** This is the Application Programming Interface(API) used to send notification messages to students.
- **Payment gateway:** It will be an API used to allow students pay their bills online. We are yet to choose which API to use. The deciding factors for the chosen API will be latency, cost effectiveness, and, ease of use.

Conclusion

The thesis is concluded with a plan of action. As discussed in the thesis, Data Flow Diagrams(DFDs) would be duly followed during development. Also, the focus will be primarily on the potential research areas pointed out in the thesis. This thesis will stand out as a base for the project’s requirements and is a proof of study done about the project before starting the development process.