Dynamic Mess Billing System using Face verification

by

Ravi Chopra

2019BCS-046

A report submitted for Summer Project

Bachelor of Technology

in

CSE

विश्वजीवनामृतं ज्ञानम्

ATAL BIHARI VAJPAYEE-

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015, MADHYA PRADESH, INDIA

Report Certificate

I hereby certify that the work, which is being presented in the report, entitled Dynamic Mess Billing System using Face verification, **Bachelor of Technology**
in **CSE** and submitted to the institution is an authentic record of my/our own work carried out during the period *May-2021* to *August-2021* under the supervision of Prof. Joydip Dhar Supervisor and Dr. Vinal Patel Co-Supervisor. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

____

Prof. Joydip Dhar Supervisor and Dr. Vinal Patel Co-Supervisor

Date ______________

The final copy of this report has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis is in the proper format as specified in the guideline for thesis preparation.
I declare that the work containing in this report is my own work. I understand that plagiarism is defined as any one or combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present as new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.
I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature: Chopra. [redacted]

Name: Ravi Chopra

Roll. No: 2019BCS-046

Date: August 9, 2021

Abstract

This thesis represents the design, implementation, and deployment process for the Minor Project (Bachelor of Technology in Computer Science Engineering 2019-2023). **The idea proposed is a dynamic billing system for the student mess that internally utilises face recognition.** In this thesis, we discuss the design, implementation, and deployment process for the project. The thesis also throws a shadow on challenges dealt with during the process of implementation. A clear description of the experiments conducted is laid down in the report.

Keywords: Face recognition, MERN stack, Insight Face, Computer Vision

Dedication

This project draws inspiration from existing systems available for billing systems in the student mess at different institutions in India. The other reason that dedicates us to develop this project is the current food wastage in the mess areas of our college. Mess workers have no idea of how much food will be utilised on the subsequent day, which leads to heavy wastage of food. This project will generate data that mess workers can use to get an idea of what quantity of food to prepare to minimise wastage. Many ideas have already been proposed for billing systems, but all lack at some points. Harnessing the power of deep learning, we can use face recognition for billing students on a per meal basis. However, this approach has some challenges that have been tried to be resolved in this work.

Acknowledgments

I want to extend my heartful thanks to **Prof. Joydip Dhar(Supervisor)** and **Dr Vinal Patel(Co-Supervisor)** for constantly guiding me through the project. They helped me to develop an excellent practice of reading the recent literature before pursuing the work. This practice increased my thirst for knowledge without any doubt. Their expertise in the field of Computer Vision helped me to think better and innovative. I am indebeted to all the professors for allowing us to develop an industry-grade project in these times of peril. They gave their valuable time for evaluating and give much-needed insights about the project.

Thanks to my colleague **Pranav Kotgire** and my family for their constant support.

I would also like to thank **Dr Andrew NG** for starting a MOOC specialisation in deep learning. Specialisation was one of the primary sources of my learning.

(Ravi Chopra)

Contents

Figures

Figure

Chapter 1

Introduction

This chapter presents an overview of the context as a part of the project developed in section 1.1. Section 1.2 introduces the objectives of the project. Next, section 1.3 presents the implementation workflow step by step. Finally, in section 1.4, the result of the research carried out is briefly introduced.

1.1 Context

This project is a part of B.Tech. Computer Science Engineering curriculum for the third semester. The objective is to develop and experiment with solutions to automate and dynamize the billing system in the mess areas of the institute. This project aims to replace the current billing system (semester-based) with a dynamic system that bills students per meal. We used a Face verification system to realize the above-stated objectives.

1.2 Implementation workflow

The workflow followed during the implementation is as follows:

Step 1: Data collection for Machine learning model

Step 2: Prepare an augmentation pipeline

Step 3: Train model using ArcFace loss function without modifications

Step 4: Record results

Step 5: Train model with modified loss function

Step 6: Record results

Step 7: Build API for system using Node.JS, Express.Js, Twilio API, Stripe API

Step 8: Test and document API using Postman

Step 9: Build UI for web application using React.JS

Step 10: Integrate Frontend and Backend

Step 11: Test, and document the final system.

1.3 Objectives

Face recognition module: The project's objective is to develop a robust face recognition system that can work in a mess area condition. The model is robust to minor changes in brightness, orientation and other factors. **Web application** : The other aim is to provide a web interface for students to interact with the system. It will include a payment gateway developed using Stripe API. Twilio API is used as a notifier for students and to provide authentication using OTP(one time password).

1.4 Research results

There is a slight increase in training and test accuracy after modifications. Test accuracy is also marginally better. The best part of the results is that the training time gets drastically reduced after modifications. Detailed results have been discussed further.

Chapter 2

Literature review

2.1 Background

This section of the thesis focuses on the literature review conducted before starting the development process. It includes a short write-up and a critical analysis of the previous works on the objective proposed in the thesis. There is a comparison drawn between different works based on standard benchmarks used in the industry.

2.2 Key related research

Various works were reviewed and analysed related to the objectives stated in the theses. These are listed below:

- 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)

2.3 Analysis

The above works were carefully analysed before moving forward. All the papers to be discussed propose different loss functions for the classification problem of faces into classes. They tend to minimize intra-class variations and maximize inter-class distances. All the loss functions are just small variations of softmax loss. Accordingly, a short analysis for each paper is stated below:

- **ArcFace: Additive Angular Margin Loss for Deep Face Recognition** : The idea is to 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:

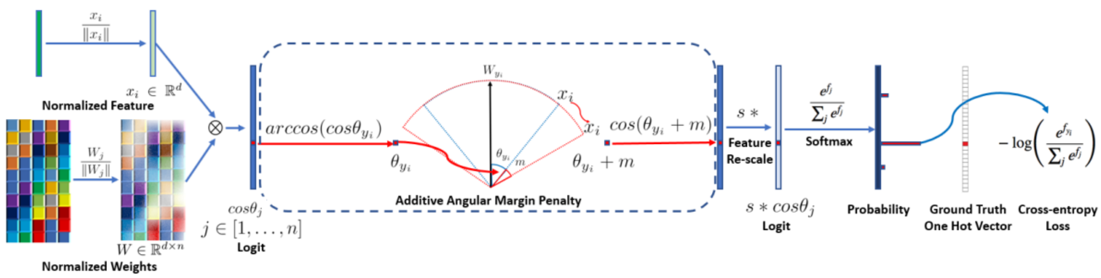

Figure 2. Training a DCNN for face recognition supervised by the ArcFace loss. Based on the feature x_i and weight W normalisation, we get the $\cos\theta_j$ (logit) for each class as $W_j^T x_i$. We calculate the $arccos\theta_{y_i}$ and get the angle between the feature x_i and the ground truth weight W_{y_i}. In fact, W_j provides a kind of centre for each class. Then, we add an angular margin penalty m on the target (ground truth) angle θ_{y_i}. After that, we calculate $\cos(\theta_{y_i}+m)$ and multiply all logits by the feature scale s. The logits then go through the softmax function and contribute to the cross entropy loss.

Figure 2.1: Source: ArcFace: Additive Angular Margin Loss for Deep Face Recognition

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{2.1}$$

- **Mis-classified Vector Guided Softmax Loss for Face Recognition** : This paper tries to design a new loss function, which explicitly indicates the hard examples as mis-classified vectors and adaptively emphasizes on them to guide the discriminative feature learning. As a consequence, our new loss also absorbs the discrimiantibility from other non-ground truth classes as well as is with adaptive margins for different classes.This paper defines a binary indicator I_k to adaptively indicate whether a sample (feature) is mis-classified by a specific classifier $\mathrm{w_k}$ (where k != y) in the current stage:

$$I_k = \begin{cases} 0, & f\left(m, \theta_{w_y,x}\right) - \cos\left(\theta_{w_k,x}\right) \geq 0 \\ 1, & f\left(m, \theta_{w_y,x}\right) - \cos\left(\theta_{w_k,x}\right) < 0 \end{cases} \tag{2.2}$$

Loss:

$$\mathcal{L}_5 = -\log \frac{e^{sf\left(m,\theta_{w_y,x}\right)}}{e^{sf\left(m,\theta_{w_y,x}\right)} + \sum_{k\neq y}^{K} h\left(t, \theta_{w_k,x}, I_k\right) e^{s\cos\left(\theta_{w_k,x}\right)}} \tag{2.3}$$

- **LinCos-Softmax: Learning Angle-Discriminative Face Representations With Linearity-Enhanced Cosine Logits** : Despite the excellent performance achieved by the angle-based softmax loss variants, one weakness is that the angle is nonlinearly mapped by a cosine function. 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. To tackle this issue, we propose a Linear-Cosine Softmax Loss to learn angle-discriminative face features more effectively. The main idea is the use of a linear-cosine logit, which is designed by performing Taylor expansion on a linear logit. This paper represents θ_j as the arccosine of $\cos\theta_\mathrm{j}$, then perform a Taylor expansion over the arccosine function, and approximate the angle using the first K terms:

  Loss:

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

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

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

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

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

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

$$P_{yi}^{LinCos} = \frac{e^{sf_{yi}\,Lin\cos}}{\sum_{j=1}^{C} e^{sf_j^{LinCos}}}$$

2.4 Research gaps

In this section we compare different loss functions which were reviewed in the previous section. The comparison is done on basis of recognition loss(in) on the basis of standard benchmarks LFW and AgeDB. It is observed that all the algorithms are on their saturation, i.e. it is quite not possible to decide the best of them.

Results: See figure 2.2

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

Figure 2.2: Results

2.5 Problem formulation

As we can see that all of the above papers already give a pretty good accuracy, it is pretty impossible to choose the best one. However, 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 approximated $\cos(\theta + \mathrm{m})$ (where m = Marginal Penalty) using the method proposed in LinCos paper. The details of experiment and results have been stated in Section Experiments and results.

2.6 Conclusion

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 remained on trying the above-proposed experiments.

Chapter 3

Methodology

This section includes a write up on tools and methods used while implementation. It lays down a clear description of the process used during development and testing.

3.1 Tools

There are variety of tools used during development of project. Major tools used have been listed below:

- **Face verification system:**
  - **Pytorch:** It is an open-source machine learning library based on the Torch library, used for computer vision and natural language processing applications, primarily developed by Facebook's AI Research lab. It is free and open-source software released under the Modified BSD license.
  - **Other libraries:** Numpy, matplotlib, MXNet, Scipy, OpenCV, PIL, Scikit.
- **Web interface:**
  - **React.JS:** React is a free and open-source front-end JavaScript library for building user interfaces or UI components.
  - **Express.JS:** Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web applications.

* **Node.JS:** Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

* **MongoDB:** MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

3.2 Workflow

The workflow followed during the implementation is as follows:

Step 1: Data collection for Machine learning model

Step 2: Prepare an augmentation pipeline

Step 3: Train model using ArcFace loss function without modifications

Step 4: Record results

Step 5: Train model with modified loss function

Step 6: Record results

Step 7: Build API for system using Node.JS, Express.JS, Twilio API, Stripe API

Step 8: Test and document API using Postman

Step 9: Build UI for web application using React.JS

Step 10: Integrate Frontend and Backend

Step 11: Test, and document the final system.

3.3 Conclusion

This section presented the workflow followed during the implementation of the system. We used Agile system of software engineering to complete the system in pre-decided timeline.

Chapter 4

Experiments and results

This section discusses the experiment conducted and the corresponding results obtained. Note: These results may vary from machine to machine.

4.1 Experiment

The idea for the experiment has been inspired from two research papers popular in the field of Face verification - ArcFace and LinCos(Cited in Bibliography). The idea is to use ArcFace but linearize cos(θ+m) using Taylor's expansion. This may reduce overfitting in the model obtained by training modified ArcFace model as per the idea laid down in LinCos paper.

4.2 Parameter settings

Here, we are linearizing cos(θ + m) with $\sum_{n=0}^{\infty}(-1)^n \frac{x^{2n}}{(2n)!}$ (Here, x= θ+m). Apart from parameters used in normal ArcFace, we can use n as another parameter. The coefficients of different terms in Taylor's expansion may be regarded as another hyper-parameters. We trained the model for n=4 and approximated cos(θ+m) with $1 - x^2/\alpha + x^4/\beta$. Ideally, α and β must be considered hyper-parameters and changed during training as per the validation loss. However, due to limitation of computational power, we directly use $\alpha = 2$ and $\beta = 33$. This guess has been made by comparing graphs of cos(θ+m) and $1 - x^2/\alpha + x^4/\beta$ for different values of α and β. Values $\alpha = 2$ and $\beta = 33$ gave the best approximation for cosine.

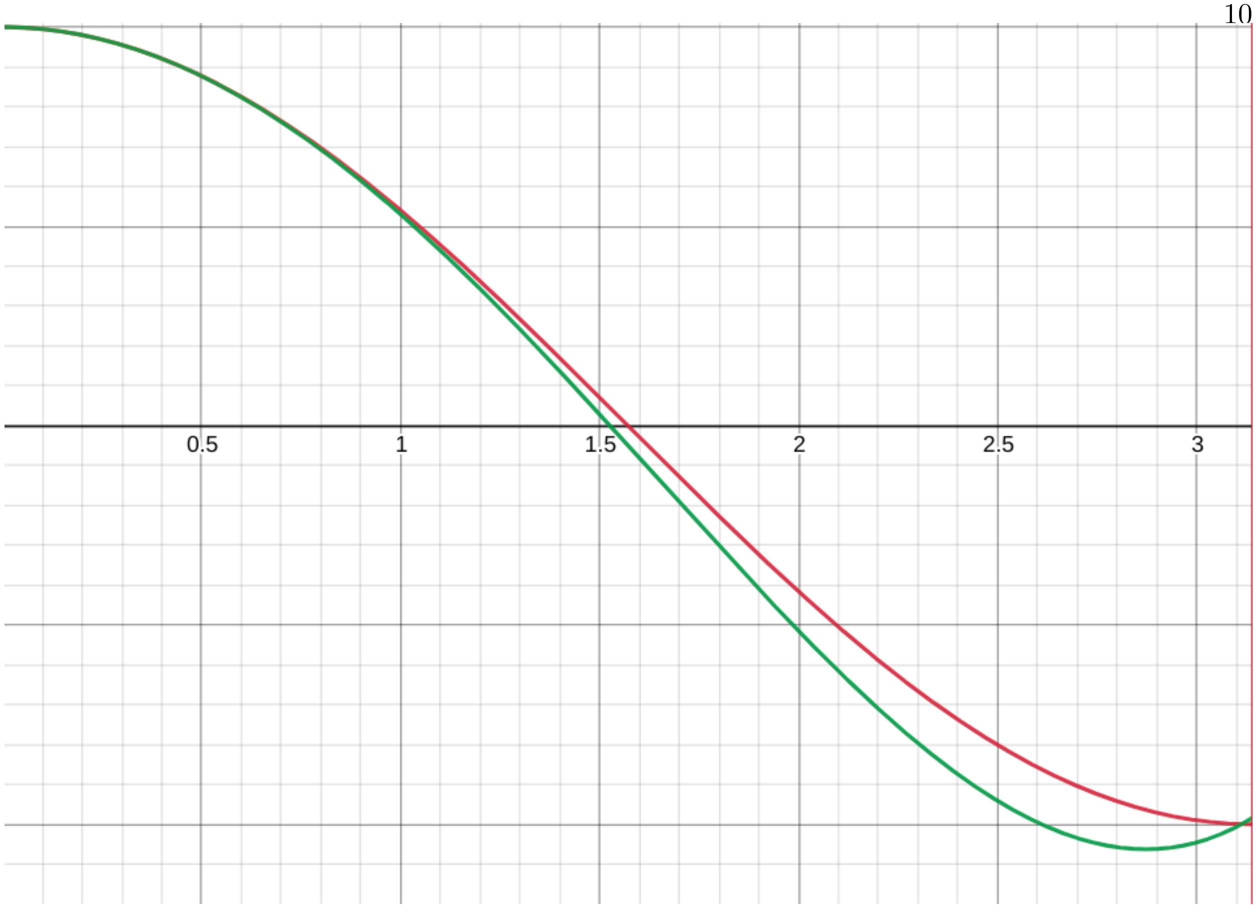

Figure 4.1: Graph: Red: cos(x) and green: $1 - x^2/2 + x^4/33$ for $0 <= x < \pi$

List of hyper-parameters and values used during training:

- n = 4 (Power of highest order term in Taylor's expansion)
- $\alpha = 2$
- $\beta = 33$
- m = 0.5 (Marginal Penalty)
- net_mode = mobilefacenet ([ir, ir_se, mobilefacenet])
- net_depth = 50
- drop_ratio = 0.6

- lr = 0.001 (Learning rate)
- momentum = 0.9
- max_epochs = 100 (Early stopping enabled)

4.3 Experiment description

First, we train the pre-trained model with ArcFace head without any modifications on the LFW dataset(Labelled Faces in the wild). Then, the pre-trained model is trained using modified ArcFace head with above-stated parameters. After that, the results of both the models are compared based on accuracy obtained on validation and test sets.

4.4 Results and discussion

The results obtained from the models have been stated in the table below:

Model	Accuracy_train	Accuracy_test	Training_time on AWS p2.xlarge instance
ArcFace	96.883%	97.0%	15.2 hours(Approx.)
ArcFace Modified	97.4%	97.683%	10.13 hours(Approx.)

*Testing has been done using LFW benchmark.

It can be seen that Accuracy increases marginally. However the point to note is that, training time reduces drastically. This may be attributed to more linearity in the model produced by Taylor's expansion. However, there is a scope of mode research to find the exact reason to this drastic decline in training time.

4.5 Conclusion

The results are not enough to conclude that modified ArcFace performs better than the original model. There needs to be more rigorous testing.

Figure 4.2: Model testing for varying brightness and orientation

Chapter 5

Discussions and conclusion

In this chapter, the work is concluded and future plan is presented. Next, the research contribution are presented. Finally, limitation of the work and possible future extensions are described respectively.

5.1 Contributions

The work presented in this paper is a step forward to a new way of thinking in Face verification. The approach of the work is to merge two existing novel ideas presented by papers- ArcFace and LinCos. It also uses augmentation techniques to improve the facebank and let model give more true_positives with respect to challenges like brightness, orientation, etc.

5.2 Limitations

The results presented in the paper are not enough to conclude anything. Therefore, there is a need for more rigorous testing and training on larger datasets. This requires for more extensive computational resources. Nevertheless, the training time drastically decreases. But the exact reason for the same has to be researched.

5.3 Future scope

There are some questions that need to be answered in the thesis. More research must be done on the idea presented in the paper using larger datasets and better computational resources.

Work is in progress to explore the idea more and improve the model.

Bibliography

[1] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[2] Wei-Feng Ou, Lai-Man Po, Chang Zhou, Yu-Jia Zhang, Li-Tong Feng, Yasar Abbas Ur Rehman, and Yu-Zhi Zhao. Lincos-softmax: Learning angle-discriminative face representations with linearity-enhanced cosine logits. IEEE Access, 8:109758–109769, 2020.

[3] TreB1eN. Insightface_pytorch. `https://github.com/TreB1eN/InsightFace_Pytorch`, 2019.

[4] Xiaobo Wang, Shifeng Zhang, Shuo Wang, Tianyu Fu, Hailin Shi, and Tao Mei. Mis-classified vector guided softmax loss for face recognition, 2019.