Comparing CORAL Loss with Cross-Entropy Loss for Age Estimation on AgeDB

Sanaulla Haq
sanaulla.haq@northsouth.edu,
Mohammed Rakib
mohammed.rakib@northsouth.edu,
Amrijit Biswas
amrijit.biswas@northsouth.edu
Iftekhar Ahmed Uday
iftekhar.uday@northsouth.edu

ECE Department
North South University
Dhaka, Bangladesh

***Abstract*—Age estimation is the process of identifying the age of a person from an image. In this paper, we perform age estimation on the AgeDB dataset by fine-tuning pre-trained ResNet models on the UTKFace dataset. We successfully break the SOTA MAE score of 13.1 years achieved by the DEX model on AgeDB. Our Resnet-152 model achieves an MAE score of 9.07 years which is a significant improvement in this field. Besides, our model is finetuned on the UTKFace dataset which is 20 times smaller than the IMDB-WIKI dataset on which the DEX model was finetuned. Moreover, we show that the CORAL framework performs better on the AgeDB dataset than the conventional categorical cross-entropy loss function. In addition, our experiments show the regular ResNet models tend to outperform the Central Difference Convolutional Networks (CDCN) for age estimation tasks**

I. Introduction

The task of age estimation from a given image has always been a challenging field of computer vision. Age estimation has many applications like security, investigation, age-based classification, surveillance, etc. There are many challenges when it comes to finding out the age of a person from images namely, gender, facial expression, ethnicity, angle of the image taken, and light. In recent years the rapid development of machine learning and neural networks have made it possible to extract meaningful information from images with relative ease. Also, the many image classifier models have had commendable performance when it comes to estimating age.

To train the models, we also need sufficiently large datasets. Some of these well-known datasets are AgeDB, UTKFace, AFAD, MORPH, CACD, etc. Deep learning algorithms can automatically find out patterns and features from these datasets and use these features to train the algorithm.

Our goal in this work is to use Convolutional Neural Networks to improve upon the MAE achieved by the DEX model in AgeDB: the first manually collected, in-the-wild age database [1] and make a comparison between Cross-Entropy Loss and CORAL Loss. Our CNN models used consisted of the ResNet34, ResNet34 pre-trained on ImageNet, ResNet152 pre-trained on ImageNet, and Central Difference Convolutional Networks or CDCN. For label encoding, in this paper one hot encoding method has been used for Cross-Entropy Loss and ordinal scale encoding for CORAL Loss. In this paper, the UTKFace dataset [5] has been used to train our model and then evaluated on the AgeDB dataset.

Concretely, we make the following contributions:

1) Our fine-tuned model achieves an MAE score of 9.07 on AgeDB for age estimation which outperforms the SOTA score of 13.1 achieved by the DEX model.
2) Our experiment results demonstrate that CORAL loss tends to always perform better than cross-entropy loss for AgeDB.
3) Our model was fine-tuned for age estimation on 20 times less samples than the DEX model and still achieves superior performance.
4) Our experiments also show that a normal ResNet model always gives better results compared to Central Difference Convolution Networks (CDCN).

The remainder of this paper is organized as follows. Section 2 provides information on the related works regarding age estimation. Section 3 describes the methods used in our project along with the model architectures and loss functions. Section 4 discusses the experiments performed and their results. Section 5 concludes the paper.

II. Literature Review

A. CORAL

For a regular classification problem, cross-entropy is used as a loss function. But the problem is that in many real-world classification problems, class labels include information about the relative ordering between labels and cross-entropy cannot capture that information. This problem usually happens when the output labels have an ordinal characteristic. For solving this problem the ordinal regression framework was being introduced but the problem was ordinal regression transformed the ordinal task into different binary classification subtasks and that caused inconsistencies among those binary classifiers. This paper Rank consistent ordinal regression for neural networks with application to age estimation [2] offered a new loss function called Consistent Rank Logits (CORAL) loss. According to the paper, CORAL provides an improvement in binary classifier consistency on different age estimation datasets without increasing training complexity. The paper used the MORPH-2, CACD, AFAD datasets. CE-CNN, CORAL-CNN, OR-CNN methods are used over those datasets. Both CORAL-CNN, OR-CNN beats the overall standard cross-entropy loss (CE-CNN). Performance improvement of CORAL-CNN over OR-CNN is shown by repeating each experiment three times using different random seeds. So this paper has shown the superiority of CORAL-CNN over all the datasets. Also according to this paper lighting condition, viewing angle, quality of the image used for training, and range of age can be factors for performance measurement.

B. *AgeDB*

In the sector of image-related work, the dataset is a significant factor. Image-related work is increasing day by day. As a result, some “in wild dataset” became available so that age attributes became available. But the problem was the datasets were collected semi-automatically and annotated. As a result, these datasets contained noise. For solving this problem, this paper [1] presented the AgeDB image dataset. First “in the wild” dataset to collect images manually. Images are annotated with accurate to the year and noise-free labels. This dataset contains 16,488 images and those images are captured under real-world conditions. For that reason, the images contain different poses, noise, and various expressions. In AgeDB, the age range is between 1 to 101 and there is no image for age 2. The ages are distributed into 10 age groups. 0-3, 4-7, 8-15, 16-20, 21-30, 31-40, 41-50, 51-60, 61-70 and 71-100. The current state of the art (SOTA) on AgeDB in terms of mean absolute error (MAE) is 13.1 years obtained by pre-trained DEX deep networks.

C. *DEX*

The DEX the Deep Expectation of apparent age from a single image was implemented for tackling the estimation of apparent age [1]. DEX uses VGG-16 architecture and is pre-trained on ImageNet contains 14M images for classification problems and is fine-tuned on the IMDB-WIKI dataset contains 500k+ images. The DEX model first detects the image from the test set. Then find out the CNN prediction from a compilation of 20 networks that comes from the cropped face. The model is the winner of ChaLearn LAP 2015. The DEX model converted the age regression problem into a deep classification problem where a softmax is used for refinement. This model uses the shelf face detector of [4] for obtaining the location of a face. When the images are not straight, the model runs the face detector on the rotated versions between -60° and 60° in 5° steps. For some 90° rotated images, the face detector uses -90°, 90°, and 180°. In the DEX model, there are less than 0.2% images from where the face detector cannot find the face. Finally, this model uses a 256×256 pixels image as input. When the model is used for a regression problem the output layer is replaced with a single neuron and for classification problems, the output layer is modified according to the number of output classes required. DEX estimated the age estimation as a piecewise regression or discrete classification with multiple discrete value labels. Also mentioned that performance can be boosted by increasing the number of classes heavily. For the ChaLearn LAP 2015 competition, DEX model uses the IMDB-WIKI dataset as training also and then used the ChaLearn dataset for Fine-tuning. DEX model also achieved the current state of the art on AgeDB of 13.1 years

III. Methodology

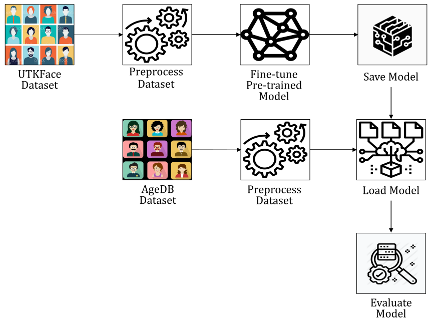

Fig. 1. Workflow Diagram

The Workflow diagram Fig. 1 shows the entire workflow of this paper. This paper uses two entirely different image datasets for train and test respectively. As shown in Fig. 1 we first take the CSV file of the UTKFace dataset where images names and images labels are stored. In the preprocessing part, specially preprocess has been done for the image labels more specifically encode the image labels. With the images and encoded image labels, fine-tuning a pre-trained model is a very heavy task. Since in this paper training and testing datasets are entirely different our motive was to overfit the pre-trained model with the UTKFace dataset. That’s why we run our pre-trained model for 200 epochs without any Learning Rate Scheduler. Initially, the StepLR scheduler was implemented but was not able to produce any satisfactory result. We saved the models in every epoch from epoch 1 to epoch 200 in the hope that since our test dataset is entirely different any epoch can perform better on the test dataset over other epochs. Before evaluating the 200 models the same preprocessing has been done for the AgeDB dataset as well. Load the models one by one and evaluate them with the AgeDB dataset. Different models produce different results over different epochs. Results will be elaborately explained in the Experiments and Results section.

A. *Model Architecture*

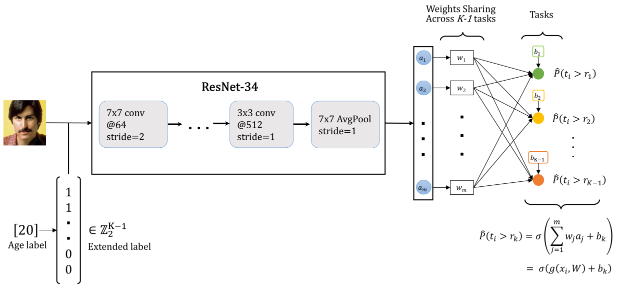

Fig. 2. Workflow Diagram

ResNet or residual network was first introduced in this paper [6]. This introduces five types of residual network bases on the internal model layer. One of them is ResNet34. Resnet34 is a 34 layer convolutional neural network that can be utilized as a state-of-the-art image classification model. This is a pre-trained model on the ImageNet dataset that has 14M+ images across 1000 different classes. Moreover, it is different from traditional neural networks because it takes residuals from each layer and uses them in the subsequent connected layers and that's why it is called residual network.

In this paper, we modified the ResNet model a little bit according to our requirements. We added a linear bias layer at the end of the model and projected the penultimate layer from 512 vector dimension to 1 vector dimension. Lastly we summed the logit values from the penultimate layer and linear bias layer as shown in Fig-2.

B. *Loss Functions*

In this section, we discuss the two-loss functions used for our experiments. One is the categorical cross-entropy loss which is normally used for multi class classification problems. The other one is the CORAL loss which is useful for ordinal data.

1) Categorical Cross-entropy Loss: Cross-entropy is usually defined as the measure of the difference between two probability distributions for a set of given random variables. Categorical cross-entropy is a specific type of cross-entropy which is used for multiclass classification problems where the target is one-hot encoded. It is calculated using Eq. 1. We will be using categorical cross-entropy since we are working with datasets that have more than two classes. The labels of our dataset are one-hot encoded. The equations for categorical cross-entropy are as follows:

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} t_{i,j}\log(p_{i,j}) \tag{1}$$

where, N = total no. of samples,
K = total no. of classes,
$t_{i,j}$ = target label of sample i for class j,
$p_{i,j}$ = predicted probability that sample i belongs to class j

Since the labels of our dataset are one-hot encoded, the CCE loss function only takes into account the loss of the one-hot class. It doesn't consider the cost of predicting other classes with high confidence. For example: suppose there are three classes: A, B, and F representing the grades of a student where A¿B¿F and our input is grade A. Now, if our model outputs a score as 10% A, 80% B, and 10% F, then we will only consider how badly it misclassified the grade A. We don't care how confident it is in predicting grade B simply because the ground truth class is not grade B. In this way, categorical cross-entropy works where it only penalizes the model based on how bad it predicted the ground truth class. Moreover, in the case of CCE, the cost of misclassifying grade A for B is the same as misclassifying grade A for F. CCE doesn't take any ordering or ranks into account which is generally required for ordinal data.

2) CORAL Framework with Binary Cross-Entropy: The CORAL framework consists of a binary cross-entropy loss function where the final layer of a model is modified to perform rank consistent multiclass classification with ordinal label encodings[Fig-2]. We are using the CORAL loss function for multiclass ordinal regression where we rank the ages of our dataset. CORAL loss allows the model to learn the ranks instead of learning to classify. To define the equation of CORAL loss we need to predefine a few things first.

Let, a dataset D = x_i, $t_{ii=1}^{N}$ be the training dataset consisting of N training examples. Here, $x_i \epsilon$ X denotes the i^{th} training example and t_i the corresponding rank, where $t_i \epsilon$ T = r_1, r_2, ..., r_K with ordered rank $r_K > r_{K-1} > \ldots > r_1$. In our case, these ranks (r_1 to r_K) are ages ranging from 1 year to 101 years. Before training, rank t_i is extended into K-1 binary labels t_1^{1}, ..., t_i^{K-1} such that t_i^{K} ϵ 0,1 indicates whether t_i exceeds rank r_K. For our case, t_i^{K}=1 if $t_i > r_K$ and 0 otherwise. For instance, if the age of an i^{th} sample is t_i = 3 years, then while extending to binary labels, t_i^{1} and t_i^{2} will be 1's as t_i is greater than r_1, and r_2 and the remaining will be zeros up to t_i^{K-1} as t_i=r_3 and $t_i < r_4$ to r_{K-1}. So, in this way, extended labels are used to train a CNN model with K-1 binary classifiers in the output layer as shown in Fig-2. Based on the binary task responses, the predicted rank label for an input x_i is obtained via h(x_i) = r_q . The rank index q is given by

$$q = 1 + \sum_{K=1}^{K} f_k(x_i) \tag{2}$$

where $f_K(x_i)$ ϵ 0,1 is the prediction of the k^{th} binary classifier in the output layer. For rank prediction, the binary labels are obtained via

$$f_k(x_i) = 1 \ \ if \ \ [p(t_i^k = 1) > 0.5] \ \ else \ \ 0 \tag{3}$$

Now, we don't want a k^{th} binary task ($f_K(x_i)$) to predict the age of a person to be more than 20 if a previous task predicted the age as less than 10. This inconsistency would be sub optimal when K-1 task predictions are combined to obtain the predicted age using Eq. 3. To avoid this problem, the K-1 (K is the no. of classes) binary tasks share the same weights but independent bias units to achieve rank-monotonicity and guarantee binary-classifier consistency [2].

The equation to calculate CORAL loss is as follows:

$$L(w,b) = -\sum_{i=1}^{N}\sum_{K=1}^{K-1}[t_i^k \log(\sigma(g(x_i,w)+b_k))+ \\ (1-t_i^k)\log(1-\sigma(g(x_i,w)+b_k))] \quad (4)$$

where, K = total no. of classes, N = total no. of examples, t_i^K = extended target label of sample i for class k, σ = sigmoid function where, $\sigma \times (z) = \frac{1}{1+\exp(-z)}$ W = weights of a neural network except the bias units of the final layer, $g(x_i,W)$ = output of the penultimate layer for i^{th} input sample x_i

IV. Experiments and Results

A. Datasets

We have used two datasets: UTKFace and AgeDB for our project. UTKFace has samples ranging from 1 to 116 years. On the other hand, AgeDB has an age span of 1 to 101 years. UTKFace is used for training raw models and fine-tuning pre-trained models whereas AgeDB is used to evaluate these models. However, both of these datasets have missing classes [ages]. Samples for ages 94, 97, and 98 are missing for UTKFace, and samples for age 2 are missing for AgeDB. Since we are evaluating our models on AgeDB, we have omitted classes 102 to 116 of UTKFace. Furthermore, as a preprocessing step, each of the datasets was cropped and aligned to feed it as input to the model. Finally, after dataset preprocessing, we used 23k+ images of UTKFace to train or fine-tune various models and their performances were assessed using all of the 16k samples of AgeDB

B. Evaluation Metrics

For evaluating all our models we have used the Mean Absolute Error (MAE), which is defined as the average of the absolute difference between the ground truth age and the predicted age. The MAE is given as:

$$\frac{1}{N}\sum_{i=1}^{N}|t_i - h(x_i)| \quad (5)$$

where, t_i = ground truth rank for sampe i, $h(x_i)$ = predicted rank for sample i, N = total no. of examples

C. CCE Based Models

In this paper, we have used pre-trained ResNet-34 and ResNet-152 as our models for fine-tuning with CCE loss function. However, due to missing classes, we have decided to perform two types of experiments. For the first type, we removed the four missing classes:- age 2 from our train set (UTKFace) and ages 94, 97 & 98 from our test set (AgeDB). So, there are a total of 97 classes in both datasets. For the other type, we kept all the 101 classes for both training and testing even if they had no samples. So essentially while evaluating models based on 101 classes, our model will predict ages 94, 97, and 98 without being trained on these classes. Observing the results of ResNet-34 in Table-1, we see that for 97 classes we achieve an MAE score of 14.12 years and for 101 classes we achieve an MAE score of 13.91 years. We see that the results are slightly better on 101 classes than 97 classes. This might be because the 101-class model was trained on more samples. Apart from that, we also trained ResNet-152 on 101 classes and this by far got us the best result with an MAE score of 13.02 years beating the SOTA result of 13.1 years [1] of the DEX model.

Table-1: Comparison of MAE scores between CORAL framework and CCE loss function using pre-trained ResNet models

Loss	ResNet-34*		ResNet-152*
	Class-97	Class-101	Class-101
CE	14.12	13.91	13.02
CORAL	9.39	9.32	9.07

*Pre Trained on ImageNet

D. CORAL Based Models

In this paper, we also performed the same two types of experiments used in CCE for the CORAL-based models. Observing the results of pre-trained ResNet-34 from Table-1, we see that for 97 classes we achieve an MAE score of 9.39 years and for 101 classes we achieve an MAE score of 9.32 years. Here also we see a similar trend of the 101 class model slightly outperforming the 97 class model. We believe this might be due to the increased no. of samples while training the 101 class model. Besides, we have also trained ResNet-152 on 101 classes and achieved an MAE score of 9.07 years which not only comfortably beats the SOTA score of 13.1 years but also surpasses the models trained with CCE loss function by a healthy margin.

E. Comparison between CCE and CORAL Based Models

Analyzing the results in Table-1 we see that, models using the CORAL framework always tend to outperform models with CCE loss function. This is because age estimation is an ordinal classification problem where ordering or ranking plays a crucial role. Using one-hot encoding for the labels with CCE doesn't take this ranking of labels into account. As a result, the cost for misclassifying an image of age 20 as 60 is the same as misclassifying it as 19 although, age 19 is much closer to 20. However, with ordinal encoding and the CORAL framework, we see that the cost varies with how far off the model predicted the label. If the prediction is close to the ground truth the cost is less compared to if the prediction is far off from the ground truth. For this reason, the CORAL framework brings about significant improvement in the performance and we see that the mean absolute error is about 4-5 years less for CORAL-based models than models that use CCE.

Table-2: MAE scores of ResNet-34 and CDCN on AgeDB

Model	CORAL
	Class-101
ResNet-34	13.97
CDCN	14.17

F. Comparing Performance of CDCN with ResNet

CDCNs can capture intrinsically detailed patterns combining both intensity and gradient information [3]. For this reason, we have tried using Central Difference Convolutional Networks (CDCN) and compared it with ResNet-34 to see which one performs better on AgeDB. To keep things simple and fair, we have used the non-pretrained version of ResNet-34 to compare with CDCN and trained each one for a total of 65 epochs on 101 classes UTKFace. It is important to note that both models used the CORAL framework while training. Upon evaluating on 101 classes of AgeDB, we see that ResNet34 performs slightly better than CDCN with an MAE score of 13.97 years compared to CDCN's 14.17 years as shown in Table-2. Moreover, CDCNs take about 4 times more training time than ResNet models. So, all in all, we see that regular ResNets are more resource-efficient and may perform better for age estimation tasks compared to CDCNs

V. Conclusion

In this paper, we have successfully achieved an MAE score of 9.07 years on AgeDB which comfortably surpasses the SOTA score of 13.1 years achieved by the DEX model. Moreover, we achieved this score on a ResNet model which was fine-tuned on 20 times less samples than the DEX model. Besides, we have also shown that the CORAL framework tends to perform better than CCE loss for AgeDB. Finally, our experiments demonstrate that raw ResNet models tend to outperform CDCNs not only in terms of performance on AgeDB but also training time.

References

[1] Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., and Zafeiriou, S. (2017). AgeDB: The First Manually Collected, In-the-Wild Age Database. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (Vol. 2017-July, pp. 1997–2005). IEEE Computer Society. https://doi.org/10.1109/CVPRW.2017.250

[2] Cao, W., Mirjalili, V., and Raschka, S. (2020). Rank consistent ordinal regression for neural networks with application to age estimation. Pattern Recognition Letters, 140, 325–331. https://doi.org/10.1016/j.patrec.2020.11.008

[3] Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., … Zhao, G. (2020). Searching central difference convolutional networks for face anti-spoofing. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 5294–5304). IEEE Computer Society. https://doi.org/10.1109/CVPR42600.2020.00534

[4] Mathias, M., Benenson, R., Pedersoli, M., and Van Gool, L. (2014). Face detection without bells and whistles. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 8692 LNCS, pp. 720–735). Springer Verlag. https://doi.org/10.1007/978-3-319-10593-2-47

[5] Zhang, Z., Song, Y., and Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. In Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 (Vol. 2017-January, pp. 4352–4360). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CVPR.2017.463

[6] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Vol. 2016-December, pp. 770–778). IEEE Computer Society. https://doi.org/10.1109/CVPR.2016.90