Ear Recognition with Convolutional Neural Networks

Assignment #3

Image Based Biometry 2020/21, Faculty of Computer and Information Science, University of Ljubljana

Urša Zrimšek

***Abstract*—Identification using person's ear is an under-used way of recognition, but it has a potential to be used in real world applications. In this report we present and analyze how ear identification could be done with use of convolutional neural networks and reach quite good results (53.5% for person identification), especially on predicting ethnicity (79.5%) and gender (96%).**

I. Introduction

This is a report of third assignment for the course Image Based Biometry. In it we explain our methodology of ear recognition using CNNs. We are only solving the last steps of a biometric pipeline, since we take already cropped ear photos and leave out the detection step, so we can better analyze the accuracy of recognition as it is not dependent on the quality of detection.

Data preparation, training commands and testing are additionally explained in my GitHub repository, where you can also find the best models mentioned in this report.

II. Methodology

In this section we will first look at our data and how we organized it, present the architectures we used and explain how we trained them.

A. Data

For training and testing we used *AWE* dataset, consisting of 1000 precropped pictures of ears from 100 people (10 each). The dataset also contains annotations with gender and ethnicity, so we separated the data in three different ways - for recognizing identity (100 classes), ethnicity (7 classes) and gender (2 classes). Of each person we selected 6 photos for training, 2 for validation and 2 for testing. Photos have different sizes, so we need to resize them to 244×244 as this is the input for the selected architectures. We also need to normalize the photos with mean and standard deviation of images from ImageNet [1] (why is explained in the next chapter).

B. Architectures

At the beginning we chose 4 different architectures (ResNet18, ResNet34 [2], DenseNet121 and DenseNet169 [3]), but we quickly saw that ResNet performs much better 1, so from now on we will analyze only first two architectures. Here we must also mention that it could be the case that DenseNet performed worse because of maximum possible batch size on 2GB GPU is 8 and 6 for the last two architectures, and it is possible that it would otherwise perform better.

Residual networks (ResNet's) are a type of deep convolutional neural networks that can have many layers, and are introducing shortcuts between layers, to help with overcoming the loss of information through them [2]. We are using ResNet with 18 and 34 layers. Because we have a relatively small dataset, we are using a model from `pytorch` that have been pretrained on 1000 classes ImageNet dataset [1]. That helps us because instead of randomly choosing initial weights, we begin the training with a net where convolutional layers already know how to detect basic shapes, we just need to finetune the model for our specific task. To be able to do that, we only need to change the last layer of the net from having 1000 neurons (number of classes on ImageNet) to the number of classes for our problem (2, 7 or 100 for gender, ethnicity and identity). We trained the network with Adam optimization and cross entropy loss criterion. During training, we saved the model that had the highest validation accuracy, so we could set the number of epochs to 50 and didn't worry about over-fitting. We still needed to set the learning rate and the size of the batch. We did that empirically, you can see the plots on which we were choosing the parameters in this notebook.

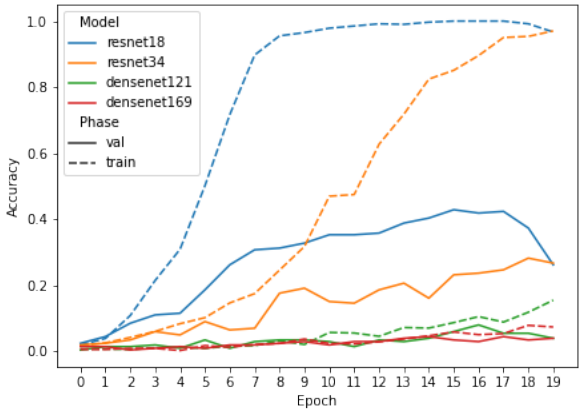

Figure 1. **Comparison of all 4 models at identity recognition training.** On the plot we can see training and validation accuracy after each epoch. We can see that DenseNet models learn much worse, and since we don't have much option to optimize it (limited batch size due to low memory on GPU), we will only optimize ResNets.

III. Results

In this section we will look at the results of some of the models that proved to learn the best on each problem. We chose the best models based on the training and validation results shown in the previously mentioned notebook.

A. Identity recognition

For identity recognition we will tested ResNet18, trained with batch sizes 20 and 15 with learning rate 0.001 and ResNet34 trained with batch size 20 and learning rate 0.001 and 0.0001. All our classes are represented equally, so a random classifier would have 1% chance of succeeding. In table I we can see that our classifiers were much better than that.

B. Ethnicity recognition

The best models for ethnicity recognition were ResNet18 trained with batch size 15 and learning rate 0.001 that reached

Table I
ACCURACIES OF DIFFERENT MODELS AT RECOGNIZING IDENTITY.

Model	Accuracy (%)
ResNet18 - batch 20	35.86
ResNet18 - batch 15	53.54
ResNet34 - lr 0.001	34.85
ResNet34 - lr 0.001	29.29

78%. Same net with batch size 20 and learning rate 0.0001 reached 79.5% accuracy. We can see confusion matrix of predictions of the second one on figure 2. On the figure we can see that the number of cases in our test classes is not at all equal. If we observe our training images, we can also see that more than half images correspond to first class. That is why we must compare our predictions with predictions of majority classifier. If we would always predict the photo belongs to first class, we would be 61% accurate, which means that our models are an improvement.

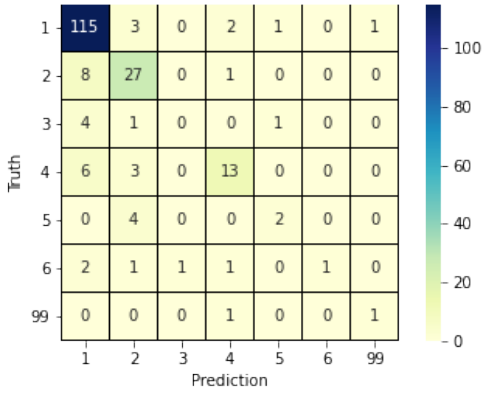

Figure 2. **Confusion matrix**. On the diagonal we can see true predictions for all classes, and out of the diagonal we can observe what kind of mistakes our algorithm did.

C. Gender recognition

When testing the models we reached the best accuracy of 96% on ResNet18 and ResNet34, both trained with batch size 20 and learning rate 0.0001. But again, we have very imbalanced classes, with 91% of males. That means that our classifier is still better than majority classifier, and that ResNet34 is better, because it correctly classified 14 females, compared to only 12 with ResNet18. We can see the number of correct and wrong predictions in table II.

Table II
PREDICTIONS OF RESNET34 AND RESNET18.

	M	F	M	F
M	178	4	180	2
F	4	14	6	12

At the end, let's look at the training of the three selected models - figure 3, from where we can see on which epoch the final model was selected - the one with the highest validation accuracy.

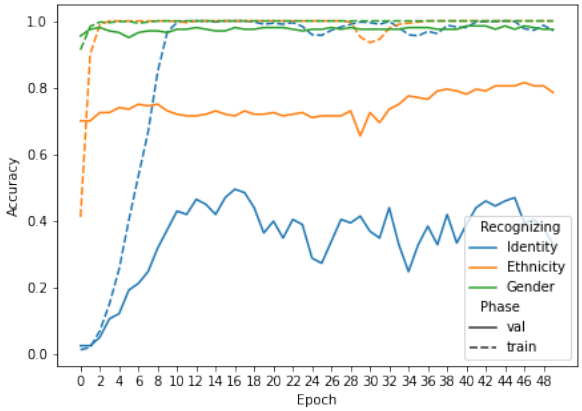

Figure 3. Training of the models that performed best on test data.

IV. CONCLUSION

We reached quite good results, specially on so small dataset. We could do better with bigger training set, specially more balanced one. As we can see on figure 2 our model is biased to predicting class 1, and would probably perform much worse if our test data would be more balanced. We could fix that by adding more photos of minority ethnicities or by artificially creating more such data. We could also append weights to classes during training, but here we knew that our test data corresponds to our train data so we left it as it is, which could leave to positive bias.

We could probably also improve our classification by choosing more complex models and train them on better machines. But here we could also come to over fitting, since we already reached almost full accuracy on training data. Which brings us again to the first problem - for better classification we would need a bigger dataset.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.