Guess The Age Contest: Group 01

Amato Mario, Avitabile Margherita, Battipaglia Lucia and Francesco Sonnessa

{m.amato72, m.avitabile6, l.battipaglia6, f.sonnessa}@studenti.unisa.it

1. Introduction

The goal for the contest is to study and propose a solution to *real age estimation*. The work we carried out consisted in evaluating whether *single expert* systems, based on DCNNs already known for the problem of *object recognitions*, can obtain good performance also for a different and complicated problem such as the one under examination. The proposed model uses a ResNet152V2 as *backbone* combined with a *fully connected* regressor to estimate the age. The tested architectures were trained using a custom loss function in order to optimize the performance of the system and the evaluation metrics of the contest.

The dataset provided is a subset of VGG-Face2 (the MIVIA Age Dataset) of 575,073 pre-cropped face images of over 9,000 identities of different ages, annotated using a *knowledge distillation* technique. The final evaluation of the models will be done on a hidden test set of 150,000 images. The evaluations proposed in this report were performed on our validation set.

The distribution of samples in the dataset contains a great imbalance between the different age groups. In fact, as shown in the graph, younger individuals (from 1 to 20 years) and older individuals (over 60) are the least represented by the samples in the dataset. This imbalance in the age estimation datasets is a known problem in the state of the art and contributes to the difficulty of the problem.

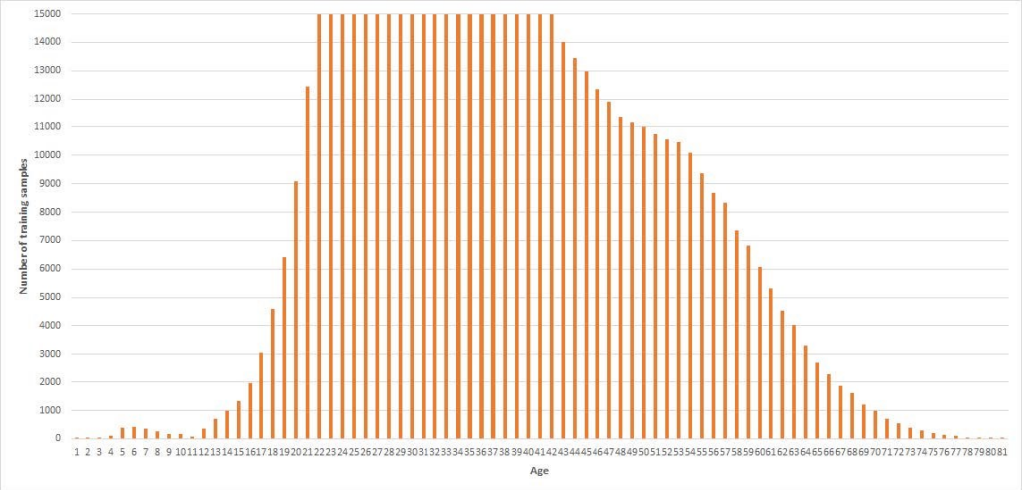

Figure 1 - Distribution of samples in the MIVIA dataset

2. Description of the method

2.1. General architecture

To solve the problem, we adopted a single expert approach, with a single DCNN as a feature extractor for the dataset images. The DCNN we have chosen is the ResNet152V2 since, from the results of the first experiments, this network reported the best performance on the evaluation metrics proposed for the contest. The ResNet152V2 is already known for having a good representation capacity and a good computational efficiency, thanks to the residual blocks and the skip connections between the convolutional layers. The architecture was completed by adding three dense layers at the top which perform age regression.

2.2. Training procedure

Dataset split. Considering the imbalance of the MIVIA dataset, rather than sampling 10% of the entire dataset, we sampled 10% for each age group, in this way we obtained a subdivision of the dataset into training and validation sets of 90% and 10% respectively.

The choice to view the dataset in this way, compared to the more common ones such as 80:20 and 70:30, was taken to guarantee a reasonable amount of data for training, considering the regression problem, and to obtain a validation set as representative as possible.

Pre-processing. As pre-processing, an online resize of the 224×224 images was made, so that they correspond to the input expected by the ResNet152V2. Also, for better model stability, the age labels have been normalized to a range of 0 to 1.

Data augmentation. For the data augumentation, a *random horizontal flip* and a *random rotation* was applied in order to simulate the possible poses in which a face can be found in a photograph. These transformations were randomly applied on the batch images when loading from disk (online transforms).

It was decided to not execute a transformation on the luminosity (*brightness*) as it is known to decrease the general performance of systems that process *Facial Soft Biometrics* and slow down their training.

For future developments, it would be interesting to observe the effects that *Face Alignment* can have on the proposed system, since, in the literature, there are conflicting opinions on the effectiveness that this transformation can have on the performance of systems for age estimation.

Training procedure and hyper-parameters. The training procedure consisted in the fine tuning of the ResNet152V2 pre-trained on ImageNet.

The training of the network took place in a way that can be defined incremental, as with the progress of the epochs the number of trainable parameters was progressively increased. The aim is to test the limits of fine tuning to understand how far the pre-trained network can perform on the new task of *age estimation*; then check if the network can get good performance even with only part of its parameters (considering it has been pre-trained).

In a first step, as a *warm-up*, only the fully connected neural regressor was trained for a short time, then also the 25% of the DCNN backbone was involved in the training. In the next two steps, the trainable backbone layers were gradually increased, going from 50% in the second phase to 100% in the third. At each step, the learning rate of the chosen optimizer, Adam, was decreased to reduce the risk of losing the knowledge previously acquired.

Below is a table that summarizes the training procedure for the various steps and hyperparameters.

Steps	Trainable	Adam learning rate	Batch size	Epochs
Step1 **(warm up)**	Only regressor	0,001	64	3
Step1	Regressor + 25% DNN	0,001	64	20
Step2	Regressor + 50% DNN	0,0001	64	30
Step3	Regressor + 100% DNN	0,00001	32	30

Table 1 – summary of the steps characterizing the training procedure

Note: the initial batch size of 64 was chosen to try to speed up the training procedure; at the same time a larger batch size may be more representative of the data distribution in the training set. In step3, to avoid exceeding the allocated memory, it was necessary to reduce the batch size to 32.

Evaluating procedure. The final evaluation of the performances was carried out by adopting the method proposed in the paper. So, a *.csv* file was generated, in which for each image in the validation set was reported in the columns: the real age and the predicted age of the evaluated image. From these data the AAR (*Age Accuracy and Regularity*) performance metric defined for the contest has been calculated and shown below.

Considering a set composed of N pairs $(y_i, \hat{y}_i)$, where y_i and $\hat{y}_i$ are respectively the real age and the estimated age for the i-th sample, the MAE (*mean absolute error*) and the *standard deviation*, σ, are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad \sigma = \sqrt{\frac{1}{8}\sum_{j=1}^{8}(MAE^j - MAE)^2}$$

where MAE^j is the mean absolute error over the j^{th} age group, 8 in total, defined by the following intervals:

$\boldsymbol{MAE^1}$	$\boldsymbol{MAE^2}$	$\boldsymbol{MAE^3}$	$\boldsymbol{MAE^4}$	$\boldsymbol{MAE^5}$	$\boldsymbol{MAE^6}$	$\boldsymbol{MAE^7}$	$\boldsymbol{MAE^8}$
$\mathbf{1-10}$	$11-20$	$21-30$	$31-40$	$41-50$	$51-60$	$61-70$	$70+$

The $mMAE$ is the measure of accuracy that takes most into account the regularity of the error and is calculated as follows:

$$mMAE = \frac{1}{8}\sum_{j=1}^{8} MAE^j$$

The final performance is expressed as a value from 0 to 10 of *the Age Accuracy and Regularity* (AAR) index:

$$AAR = \max(0; 5 - mMAE) + \max(0; 5 - \sigma)$$

3. Loss function design

The loss function used for training is a linear combination of two metrics: the $mMAE$ and the *standard deviation,* σ.

$$L_{AAR} = \alpha \cdot mMAE + \beta \cdot \sigma$$

To maximize the AAR, the goal of the network is to minimize these two measures together as much as possible. α and β are parameters of the loss function to weights the contribution of the two metrics.

In our case, not being interested in giving greater emphasis to a particular metric and in order to better conform to the AAR of the contest, we used loss parameters equal to $\alpha = \beta = 0{,}5$.

4. Experimental results

4.1. Experimental framework

The experiments conducted had the aim of determining the final architecture subsequently subjected to refinement. Below are the results of the most significant tests on the validation set.

Architecture	MAE^1	MAE^2	MAE^3	MAE^4	MAE^5	MAE^6	MAE^7	MAE^8	$MAE \downarrow$	$mMAE \downarrow$	$s.t.d. \downarrow$	$AAR \uparrow$
ResNet152V2 (v1)	8,50	3,83	2,68	3,50	3,72	3,40	3,55	3,53	3,36	4,09	**1,84**	**4,06**
ResNet152V2 (v2)	11,10	3,15	2,35	3,46	3,67	2,48	2,97	3,89	3,35	4,13	2,79	3,07
EfficientNetB5	10,50	3,45	2,39	3,97	3,18	2,41	2,85	3,76	**3,09**	**4,06**	2,68	3,26

Table 2 – test results on candidate architectures for final submission

ResNet152V2 (v1) – Architecture composed of ResNet152V2 as backbone and 3 dense layers without activation function, respectively of 1024, 512 and 1, to obtain the regressor.

ResNet152V2 (v2) – Architecture composed of ResNet152V2 as backbone, a dense layer with L2 regularization and 1024 ReLu activation function, a GlobalAvaragePooling layer, a 128 layer with ReLu activation function, a dropout layer with value 0.2, a normalization layer, 1 dense layer of 64 with ReLu activation function, and a dense layer of 1 to implement the regression.

EfficientNetB5 – Architecture composed of EfficientNetB5 as backbone, a GlobalAvaragePooling layer, a dense layer with L2 regularization and ReLu activation function of 1024, a Dropout layer with value 0.3, a dense layer with ReLu activation function of 128, a Dropout layer with value 0.1, a normalization layer, a dense layer with ReLu activation function of 64 and a dense layer of 1.

The models shown were evaluated (2.2) after step1 of the overall procedure (see Table 1) and compared in relation to the metrics and the AAR index. The model chosen was the one with the best overall metrics and the largest AAR.

For future developments, it would be interesting to try learning strategies other than the one in incremental steps presented.

4.2. Results

Once the experiments were completed, the refinement of the ResNet152V2 (v1) architecture continued. Completing the training procedure the results obtained on the validation set are:

Architecture	MAE^1	MAE^2	MAE^3	MAE^4	MAE^5	MAE^6	MAE^7	MAE^8	$MAE \downarrow$	$mMAE \downarrow$	$s.t.d. \downarrow$	$AAR \uparrow$
ResNet152V2 (v1)	5,36	2,38	1,80	2,17	2,20	2,01	2,26	2,79	2,08	2,62	1,20	6,18

Table 3 – final results

At the end of the training procedure the system achieved an AAR score of 6.18. MAE^1, for the age range [1-10], remains a problem in prediction; more encouraging are MAE^2, MAE^7 and MAE^8 of the other less represented age groups. The $s.t.d.$ obtained indicates a good uniformity of the MAE among the various age groups.

As a further analysis of the results, graphs were created on the AAR trend and on the custom loss function using the logs obtained during the training.

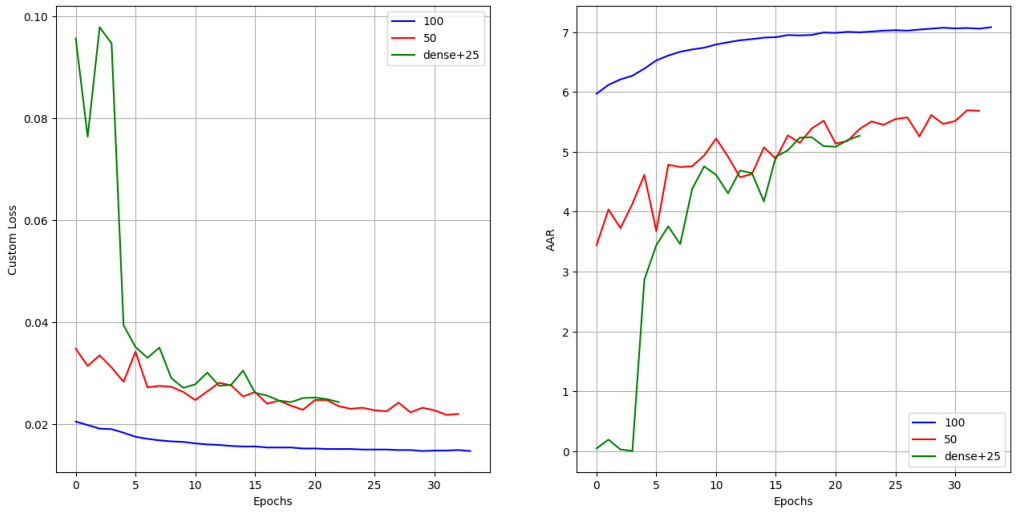

Figure 2 - AAR e custom loss function during the training

With just over 10 epochs compared to step1, the model did not achieve great improvements in step2 of the training procedure: the performances obtained with 25% of the trainable parameters were mostly equaled despite the fine tuning of step1. The reason may be the learning rate still too high at the beginning of step2 which may have inhibited what was learned in step1. The results in step 3 were substantial, which effectively decreed the final AAR.

It is interesting to note the evolution of the AAR index during the first and second steps, much more irregular (rough) than the third step (smooth): probably again due to the learning rate or the batch size of 64.

Finally, we observe the difference between the AAR calculated on the validation set during the training and that obtained from the evaluation procedure described in paragraph (2.2); the difference is about one point. The reason is due to the batch AAR evaluation performed on the validation set by the Keras framework. In fact, by averaging the AAR obtained on each batch, a higher AAR will generally be obtained, this because, on average, in a batch there will be no samples for the less significant age groups due to their small number; this implies a zero MAE on these groups and therefore a lower $mMAE$.

Below is an extract of the metrics obtained on some batches during the execution of the Keras evalute method on ResNet152V2 (v1).

```
maej: [0,    2.13, 1.33, 1.19, 0.74, 0.41, 1.01, 0]     mae: 1.89  mmae: 0.84  std: 1.24   AAR: 7.90
maej: [0,    1.22, 1.45, 1.77, 1.91, 1.10, 2.80, 0]     mae: 2.01  mmae: 1.28  std: 1.15   AAR: 7.56
maej: [0,    1.52, 0.95, 1.41, 1.47, 1.21, 1.04, 0]     mae: 1.38  mmae: 0.95  std: 0.72   AAR: 8.32
maej: [0,    0.63, 0.82, 1.32, 1.37, 0.90, 0.88, 0]     mae: 1.47  mmae: 0.73  std: 0.88   AAR: 8.38
maej: [0,    8.35, 8.61, 8.63, 8.74, 8.64, 7.94, 15.43] mae: 2.19  mmae: 8.29  std: 7.22   AAR: 0
maej: [0,    0,    1.19, 1.07, 1.32, 1.04, 0,    0]     mae: 1.05  mmae: 0.57  std: 0.75   AAR: 8.66
maej: [0,    1.09, 1.48, 1.81, 1.25, 1.06, 1.01, 0]     mae: 1.73  mmae: 0.96  std: 0.97   AAR: 8.05
maej: [1.46, 0, 1.49, 1.94, 1.73, 1.28, 1.79, 0]        mae: 1.74  mmae: 1.21  std: 0.90   AAR: 7.88
maej: [0,    4.17, 1.79, 2.06, 1.38, 1.56, 0,    2.60] mae: 2.61  mmae: 1.69  std: 1.56   AAR: 6.73
maej: [0,    0.80, 0.71, 1.48, 0.93, 1.28, 1.32, 0]     mae: 1.89  mmae: 0.81  std: 1.19   AAR: 7.98
```

Table 4 - each line shows the metrics calculated on a batch during an epoch of keras' evalueate method on ResNet152V2 (v1)

4.3. Repeatability and stability of the results

The training process was made mainly using the workstation of the department and the Kaggle online service. Kaggle was used in the first phase of the experiments to parallelize the trials on the candidate architectures.

Regarding repeatability, in the previous paragraphs we have provided an accurate description of the training process (2.2), of the model implemented (2.1) and of the evaluation procedure adopted (2.2).

We do not have sufficient data to support an accurate assessment of stability. For future developments, the behaviour of the system could be studied in contexts of noise (e.g., brightness) and with different datasets in order to evaluate its robustness and stability.

4.4. Prediction of the results on the test

It's hard to predict performance on an unknown test set. The measures on which we rely most are those obtained through the csv; therefore, if the test set is distributed like the data set provided, we should achieve at least a score equal to that obtained on the validation set.

Wanting to make some hypotheses, it is reasonable to think that if in the test set there was a greater concentration of the less represented age groups, then we could have worse performances than those obtained on our validation set; if, on the other hand, there was a greater concentration of samples from the most represented age groups in the test set, then we could even exceed the score obtained.