Guess The Age Contest: TEAM 7

Roberto Falcone, Andrea De Gruttola, Salvatore Grimaldi, Luigi Schiavone

{r.falcone13, a.degruttola, s.grimaldi29, l.schiavone7}@studenti.unisa.it

1 Introduction

Age estimation from face images is nowadays a relevant problem in the field of artificial vision, in the light of the several real applications to which it is related, such as digital signage, social robotics, business intelligence and access control. It is one of the most challenging topics in facial analysis for several reasons, namely the significant differences among faces in the same age range, a high dependency of aging traits on gender and ethnicity, the lack of huge, balanced and *"in the wild"* datasets, the difficulty of the task even for humans, the fact that image corruptions happening in real environments can be assimilated to aging factors.

During these last years, thanks to deep learning paradigm breakthrough, the number of published papers on this topic has grown enormously. In particular, the best state-of-the-art methods use ensembles of DCNNs (Deep Convolutional Neural Networks), making the obtained models not usable in real applications, since they require prohibitive computational resources, high inference times, huge training sets. Another important critical issue is that most of state-of-the-art methods are less accurate in the age estimation of children and elders, as the available datasets are unbalanced and lack of samples for these categories of people, so these models often show a poor regularity in terms of errors among age groups.

Based on what has been previously mentioned, the Department of Information and Electrical Engineering and Applied Mathematics (DIEM) of University of Salerno decided to run a contest: *"Guess The Age"*, which is an international competition among methods based on DCNNs for Age Estimation (AE) from facial images. The following report describes the solution that we, as TEAM 7, have designed and proposed: its effectiveness is demonstrated by the good results obtained on the dataset provided by contest organizers, which is one of the biggest available with exactly 575.073 images, and whose name is MIVIA Age Dataset; moreover, this dataset, just like the most popular age estimation ones, is characterised by a similar long-tailed distribution. Our solution is going to be evaluated in terms of accuracy and regularity on a test set of more than 150.000 images, different from the ones available in the training set but with a similar distribution. In particular, the adopted metric is the Age Accuracy and Regularity (AAR) index.

Our most important design choices are:

- Inspired by [10] and [3], we decouple the learning procedure into representation learning and classification, since it is one of the best known strategies to deal with long-tailed distributions. In fact, our purpose is obtaining a method as balanced as possible, keeping a high accuracy.
- The representation learning stage is intended to train a backbone network with the objective of obtaining a robust feature extractor. The classification stage, instead, is intended to train a classifier network aiming to determine the final age estimate, exploiting the previously extracted features.
- The adopted backbone is ResNet-50 pre-trained for face recognition on VGGFace2 Dataset, in fact the results of recent proposed methods suggest that age estimation pre-training is significantly more effective than the general task pre-training and the training from scratch.
- The followed approach in terms of age estimation problem formulation is Distribution Age Encoding (DAE), which has been proposed in [6] and is obtained by substituting the one-hot encoding vector with a statistical distribution centered on the estimated age. DAE paradigm generally achieves better performance with respect to the regressive paradigm and comparable with the one achieved by Classification Age Encoding (CAE) [5].
- The last layer of the classifier network is made of 100 neurons and its activation function is softmax, so the output of each neuron represents the probability that the input sample has an

age $k \in$ [1,..., 100]. In particular, both during training and at inference time, the predicted age is obtained using an expectation regression on the output distribution.

- Crucial for the achievement of our good results is the utilization of specifically designed loss functions for maximizing the AAR: during representation learning stage the adopted loss function is a linear combination of KL (Kullback-Leibler) divergence and $\ell 1$ loss, while during classification stage we use L_c, which is able to further narrow down the standard deviation of different age groups thanks to a smart introduction of Mean Squared Error (MSE).
- We use a classifier network made of three dense layers of 100 neurons each, achieving better results compared to using only one dense layer.
- We perform data augmentation by enriching the training set with variations that occur in real world: random horizontal flip, brightness and shear. This choice is significant to achieve good performance and adequate generalization ability, because it reduces the sensibility of the resulting network to translation, viewpoint and illumination.
- The training procedure is characterised by the use of SGD as optimizer, since it is the most used for AE problem, ensuring good results and not exaggerated training times (e.g. it is adopted in [8], [3], [11]); a progressive decrease of learning rate (lr) after a certain number of epochs without improvements in terms of AAR, which gives us the chance to make the most of the learning capabilities of the network; a batch size of 32, since it is a standard choice and we decide to make more effort on other design choices.

2 Description of the method

This section provides the method architecture description, an insight about the provided dataset, its division in training and validation sets, a focus on design choices in terms of pre-processing, data augmentation and a detailed explanation of the conducted training procedure.

2.1 Network architecture

The backbone of our model is represented by ResNet-50 pre-trained for face recognition on VGGFace2 Dataset [4]. We make this choice because, with the same learning procedure, pre-processing, data augmentation policy and classifier architecture, it is the backbone that obtains the best results compared to SeNet-50 and VGG-16, which are the other two possible backbone networks taken in consideration during tests. An in-depth insight into the results of the various tests carried out is reported in the section 4 of this report.

ResNet-50 stands for Residual Network and is a specific type of convolutional neural network (CNN) introduced in [9]. It is a 50-layer CNN (48 convolutional layers, one MaxPool layer, and one average pool layer) characterised by an intense use of skip connections. ResnNet-50 is an evolution of ResNet-34, which is based on the VGG neural networks (VGG-16 and VGG-19) but has fewer filters and is therefore less complex than a VGGNet.

We decide to adopt face recognition pre-training for several reasons. First of all, training from scratch a deep network like ResNet-50, VGG-16 or SeNet-50 is not trivial: because of the size of the dataset, the process may require weeks of computation even using powerful systems with GPU accelerators. This usual training procedure can be avoided by reusing the intermediate-level features of a pre-trained network. Such a technique is known as *transfer learning* or *fine tuning* and according to this paradigm very large datasets (i.e. ImageNet and VGGFace2 Dataset) need to be used only for training networks devoted to solve general problems (i.e. object recognition and face recognition) and the obtained networks are then fine tuned for specific tasks, like age estimation. Three pre-training strategies to deal with age estimation problem are known in literature: they are general task (GT), face recognition (FR) and age estimation (AE) pre-training. GT pre-training is a typical solution adopted for transfer learning and consists of a pre-training of a deep network for solving the general problem of image classification with a large number of classes. Hence, the features learnt by the hidden layers are very general and the performance on the problem at hand are typically only weakly affected by adopting the fine tuning procedure instead than a complete learning from scratch. This is the reason why we decide to utilize FR pre-training, which instead consists of a training phase by using datasets annotated with the identity of the subject. The rationale behind this choice is that the deep network learns not a generic object representation but a face representation, that is much more useful for a face analysis task such as AE. Actually, what appears to be in literature the best choice is AE pre-training [5], but we can not adopt it because of the contest limitations: for training our model we are allowed to use only the provided images.

The approach that we decide to use in terms of age estimation problem formulation is DAE, which is a modification of the CAE strategy, obtained by substituting the one-hot encoding vector with a statistical distribution centered on the estimated age. Assuming that we have an age dataset D, the ith sample in D is represented by $(x_i,\ y_i,\ z_i)$, where x_i is the input image, y_i is the provided age label and z_i denotes the computed age label distribution. In particular, the latter is obtained using a typical Gaussian distribution:

$$z_i^k = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(k-y_i)^2}{2\sigma^2}\right) \tag{1}$$

where k ϵ [1,..., 100] and standard deviation σ is set to 1.
The DAE choice is motivated by the characteristics of the loss functions adopted at representation learning stage and at classification stage: the introduction of KL divergence suggests to use a Gaussian ground-truth label distribution instead of an easy one-hot encoding vector. Though DAE approach could seem more complicated than CAE, it actually does not significantly increase the needed training efforts in terms of time and learning cycles. Moreover, the results achieved by [1], winner of the competition ChaLearn LAP 2016, and the analysis conducted in [2] suggest that DAE is the most effective paradigm for AE.

Both during training phase and at inference time, the predicted age $\hat{y}_i$ is obtained by using an expectation regression (5) on the output distribution and not just picking the age corresponding to the output neuron whose output is the largest. This choice is motivated by our use of DAE paradigm: given that the ground-truth label distribution is a Gaussian whose average value is the true age, then it is reasonable to extract the predicted age as statistical average of the prediction distribution (expectation regression).

It is necessary to state that the network trained at *representation learning stage* is ResNet-50 pre-trained for FR with attached an output dense layer of 100 neurons, whose activation function is softmax. After this stage, the output layer is removed and replaced by three dense layers (initialized with random weights) of 100 neurons each, of which the first two are characterised by *relu* as activation function (which is the default activation when developing multilayer perceptron), while the last, that corresponds to the actual output layer, has softmax as activation. These three layers correspond to our classification network (i.e. classifier), that represents the only portion of the entire network whose weights can change during *classification stage*: all the other trainable layers are frozen.

It is interesting to highlight that classifier architecture has been chosen after some tests that are detailed in section 4. In particular, for our specific problem and our specific backbone, a classifier made of three dense layers turns out to be better than one made of a single dense layer, contrary to what is reported in [10], where ResNetXt50 is adopted as backbone, the problem analysed is face recognition, and as the MLP used as classifier goes deeper, the performance get worse. The discovery is then very interesting: *when decoupling the learning procedure into representation learning and classification, the best classifier (in terms of its depth) is not always the simplest one, but depends on the specific problem, the network, and the level of discrimination among classes learned at representation stage.*

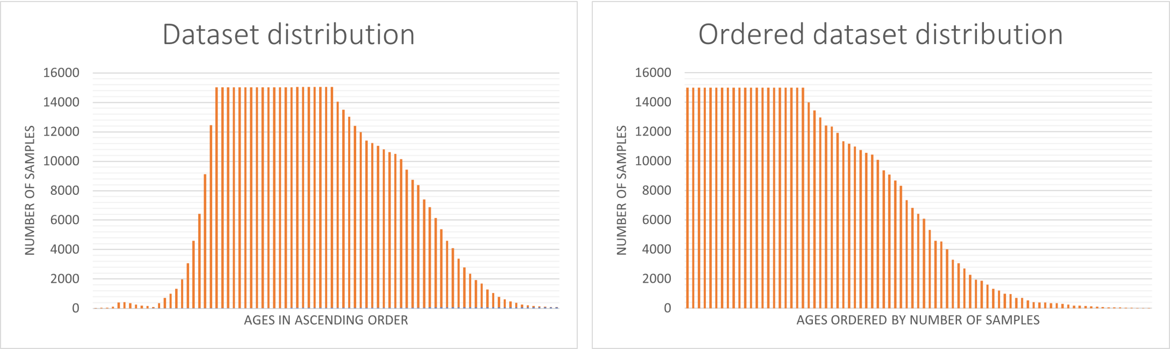
2.2 Training procedure

This subsection focuses on the statistical analysis of the provided dataset, its division in training and validation sets, and all the choices in terms of training strategies, all appropriately justified.

2.2.1 Dataset

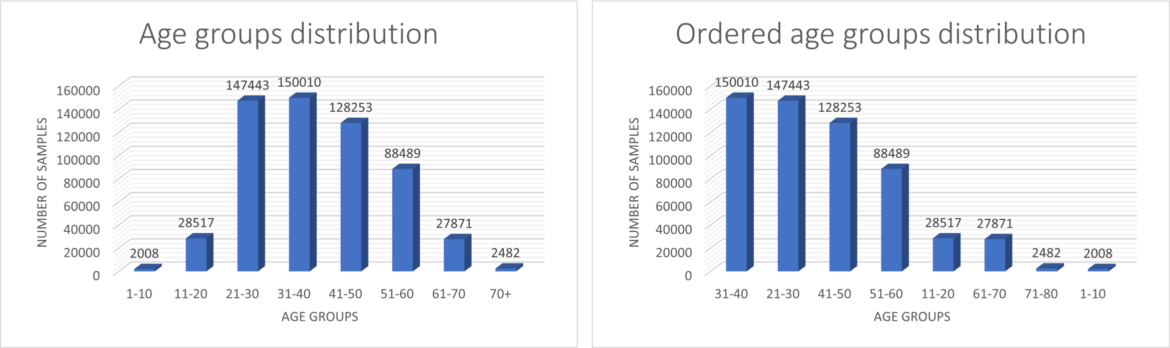
The provided dataset is called Mivia Age Dataset, includes 575.073 images of more than 9.000 identities, got at different ages, annotated with age labels and is among the biggest publicly available datasets of faces in the world with age annotations. In particular, the images have been extracted from the VGGFace2 Dataset and annotated with age by means of a knowledge distillation technique [8], making the dataset very heterogeneous in terms of face size, illumination conditions, facial pose, gender and ethnicity. Of course, this is a good starting point if we hope to obtain a model that we would like to use "in the wild", i.e. uncontrolled conditions. Each image of the dataset contains a single face, already cropped, which is relevant because it frees us from the necessity of introducing face detection inside pre-processing pipeline.

Mivia Age Dataset is characterised by a long-tailed distribution: as we can see in Figure 1b and Figure 2b, there is a very relevant imbalance in terms of representativeness of different age labels. Particularly, the instance-reach (high-frequency) classes are the ones corresponding to middle ages, while the instance-scarce (low-frequency) classes are the ones corresponding to elderly and children. Such imbalance is a recurring feature of age estimation datasets because of the apparent difficulty in finding numerous photos labelled with age of people belonging to the outermost age groups. It is interesting to observe that also for other important problems (e.g. face recognition) in the field of artificial vision, the available datasets typically present a certain degree of imbalance: *the long-tail distribution of the visual world poses great challenges for deep learning based classification models on how to handle the class imbalance problem* [10].



(a) (b)

Figure 1: Figure (a) shows the dataset samples distribution, ordered by class from the youngest to the oldest age. Figure (b) shows the same distribution but sorted by the number of samples in class. Dataset distribution is clearly an example of *long-tailed* distribution: the number of samples per class generally decreases from head to tail classes, where head-class denotes high-frequency class, and tail-class denotes low-frequency class. Moreover, it is relevant to say that Mivia Age Dataset does not present any image depicting a person older than 81, and that there are some ages (eg. 1, 2, 3, 4, 76, 77, etc.) for which the number of samples is tremendously low.



(a) (b)

Figure 2: Figure (a) shows the dataset samples distribution, grouped by age ranges. Figure (b) shows the same distribution but sorted by the number of samples in age groups. It can be immediately seen that the age groups [1-10] and [70+] are extremely poor, [11-20] and [61-70] are under-represented groups, and [21-30], [31-40], [41-50], [51-60] are very well-represented groups.

Mivia Age Dataset has been randomly shuffled and then divided into 5 disjoint subsets of similar dimensions, so that each of them contains about 20% of all the provided images. All the experiments are conducted adopting the first 4 subsets as training set and the last subset as validation set. This choice is done to obtain comparable results. It is important to say that the best architecture obtained from these experiments is then further tested with a 5-fold cross validation, utilizing the same 5 subsets depicted above.

The performance are always evaluated making use of the same metric on which our final proposed model is going to be evaluated by ”Guess The Age” organizers: AAR index. Further details about the applied 5-fold cross validation and performance evaluation are given in section 4.

2.2.2 Pre-processing and Data augmentation

Pre-processing techniques are used for making the input data suited for training a neural network. In our case, it is important to state that each image of the provided dataset contains only one face already cropped: it is not necessary to introduce a face detection mechanism inside pre-processing pipeline .

Of course, *image resizing* is instead mandatory. Given that all the tested backbones (SeNet-50, VGG-16, ResNet-50) have an image input size of 224×224, every image must be resized to these same dimensions. In particular, we use *bilinear interpolation* as interpolation method involved in image resizing. This choice is justified by the fact that bilinear interpolation guarantees to mantain the overall appearence of the image because it takes into account the color values of the pixels surrounding a given point, rather than just the color value of the closest pixel. This is relevant when dealing with face images: losing as few details as possible after resizing plays an important role in preserving the aging factors (e.g. wrinkles and other skin blemishes). Another important pre-processing operation performed on each image is *rescaling*: data normalization makes convergence faster while training the network. The applied rescaling technique sets the new range of each sub-pixel to [0-1], while the original minimum and maximum values for each sub-pixel are 0 and 255. Rescaling, in fact, helps to ensure that the data is within a short and manageable range, which can improve the performance and stability of the neural network.

Data augmentation is a technique that can significantly expand the training dataset by creating new samples through the application of modifications to the images in the original dataset. The idea behind data augmentation is that, as the size of the network increases, the curse of dimensionality requires a corresponding increase in the amount of training data in order to achieve optimal performance and generalization capabilities. Data augmentation is without doubt one of the keys of deep approaches and age estimation does not constitute an exception [5], as we can see by analysing methods that achieve state-of-the-art accuracy on most of the available datasets (e.g. [12]). Moreover, all the first three classified methods at ”Guess The Age 2021” [7], which are [3], [11] and [13], make use of a kind of data augmentation. What just said justifies our use of data augmentation. Particularly, the transformations that we choose to apply on *training samples* are the following ones:

1. *Random Horizontal Flipping.* The image is horizontally flipped with probability 1/2
2. *Random Brightness.* The factor with which image brightness is adjusted is a float randomly picked from the interval $[-0.15, 0.15]$
3. *Random Shear.* The image is sheared on X axis of a float factor randomly picked from the interval $[-0.1, 0.1]$, and is sheared on Y axis of a float factor randomly picked always from the same interval $[-0.1, 0.1]$

Note that the extremes of the intervals within which the data augmentation factors vary are empirically chosen, with the objective to get images that are representative for the dataset: the transformations are reasonable and not exaggerated. The applied data augmentation policy guarantees to improve generalization ability of the network, by making it more robust in respect to translation, viewpoint and illumination.

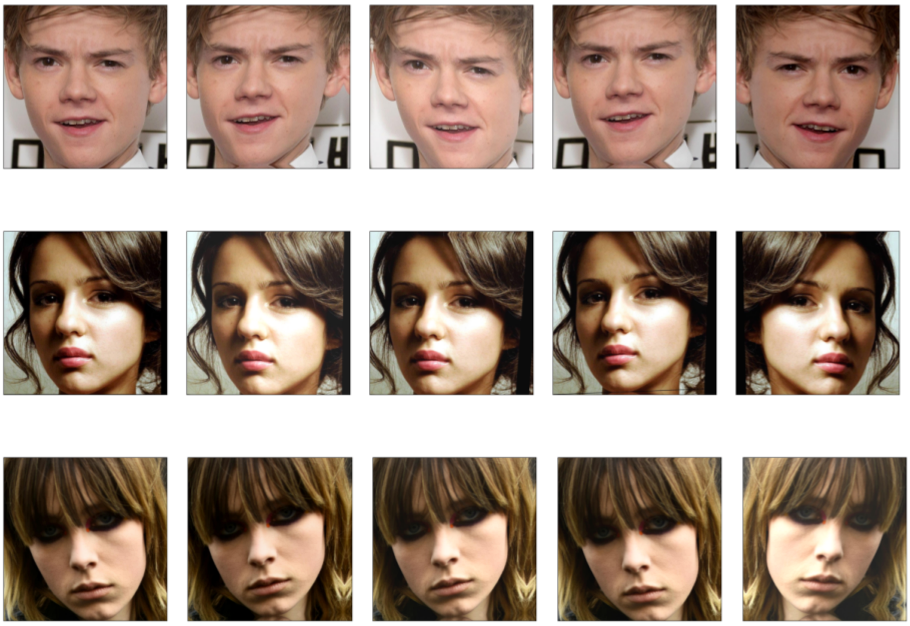


Figure 3: On each row, the leftmost image is an original sample picked from Mivia Age Dataset, while the other four images are some samples obtained by applying the data augmentation policy above described.

2.2.3 Training with "decoupling"

The success in achieving high performance is closely tied to the methods employed for training the deep network, as supported by numerous experimental data gathered in various application areas. Our training procedure design is driven by the awareness that the main objective of the contest is *achieving a small but balanced age estimation error across all the age groups* depicted in Figure 2. In fact, the main challenge is the regularization of the performance on the age groups despite the provided dataset significant imbalance. Networks trained on unbalanced data usually show poor performance in classes with fewer samples, and this is exactly what we do not want to happen in our case.

It is crucial to remember that our starting point is represented by ResNet-50 pre-trained for face recognition on VGGFace2 Dataset. Inspired by previous research in the field of long tailed-recognition [10] and its successful application to the AE problem [3], we decide to decouple the learning procedure into representation learning and classification to obtain a balanced model, able to perform well across all age groups.

- **Representation Learning Stage.** Its purpose is making the network able to learn high-quality representations, so we can obtain a robust feature extractor to distinguish different ages. The optimizer adopted is SGD (Stochastic Gradient Descent) and the batch size is 32 for the reasons already specified in Section 1. The learning rate is initialized to 0.005 and reduced with a factor of 0.2 every 4 epochs without an increase in the AAR computed on the validation set (val_AAR). Learning rate reduction is performed 3 times: after the third, if val_AAR does not increase for 4 consecutive epochs, then the training is stopped. Learning rate initial value is inspired to [8] and [3], while its reduction factor is inspired to [8], since their effectiveness. The rationale is making the most of the learning capabilities of the network, fine tuning its weights as best as possible. Early stopping is good for two reasons: it is a common form of regularization, so it allows us to reduce overfitting; it also helps us to reduce training times, which is crucial if we want to make several experiments. Fundamental to get good results is the use of (2) as loss function: its detailed description and the motivations of its application are well explained in Subsection 3.1. Note that the model returned by the representation learning stage is the one that obtains the best val_AAR.

- **Classification Stage.** After the end of the previous stage, the output layer is removed and replaced by three dense layers (initialized with random weights) of 100 neurons each, of which the first two are characterised by relu as activation function, which is the default activation when developing multilayer perceptron, while the last, that corresponds to the new actual output layer, has softmax as activation. These three layers correspond to our *classification network*, that represents the only portion of the entire network whose weights can change during classification stage: all the other trainable layers are frozen. The purpose of classification stage is re-adjusting the decision boundaries specified during representation learning. This is done thanks to an appropriate training of a classifier that exploits the representations provided by the frozen layers. Experiments shown in [10] demonstrate that a good way of training the classifier is with *class-balanced sampling*, which is the same strategy adopted in [3]. For this reason, we decide to train the classifier network guaranteeing that each class (age) has an equal probability of being selected. This can be seen as a two-stage sampling strategy, where a class is first selected uniformly from the set of classes, and an instance from that class is subsequently uniformly sampled. As previously done for representation learning stage, also for classification stage the optimizer is SGD and the batch size is 32. This choice is inspired by a similar strategy adopted in [3], where optimizer and batch size do not change from one stage to the next one. It is necessary to observe that during this stage the initial learning rate is 0.00015 instead of 0.005 (adopted for representation learning). A similar but not identical choice in terms of reduction of initial learning rate from representation learning to classification is made in [3]. We choose to start from a lr that is significantly low because with higher values (e.g. 0.001) we encounter problems in terms of network convergence, given that during training the loss spikes up instead of decreasing. Learning rate is reduced with a factor of 0.2 every 3 epochs without an increase in val_AAR. Learning rate reduction is performed 3 times: after the third, if val_AAR does not increase for 3 consecutive epochs, then the training is stopped. The reason of this lr management is identical to what has already been said for representation learning. Note that the *patience* is less than the one utilised in the previous stage (i.e. 4); in fact, the classifier network, given its reduced dimensions, converges faster, so a higher patience would be useless. Essential for achieving a good performance is the employment of (6) as loss function: Subsection 3.2 provides an explanation of the reasons behind its implementation. It is worth mentioning that the final model is the one which reaches the best val_AAR.

3 Loss function design

This section focuses on the loss functions utilised during network training. They are clearly inspired to the ones adopted in [3], in the light of their effectiveness witnessed by [7]. Since our training procedure is composed of two different stages, we need two losses, one for each stage.

3.1 Representation learning stage

For representation learning stage we use an AAR-inspired loss function L_r (representation loss), which is composed of two terms, L_{ld} and L_{er}:

$$L_r = L_{ld} + \lambda L_{er} \tag{2}$$

The first term (L_{ld}, i.e. label distribution loss) is the *KL divergence* between the true label distribution $z_i^k = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(k-y_i)^2}{2\sigma^2}}$ and the prediction distribution $\hat{z_i^k}$, given a sample x_i with true label y_i.

$$L_{ld} = \sum_{k=1}^{100} z_i^k \log \frac{z_i^k}{\hat{z_i^k}} \tag{3}$$

Note that divergence is a measure of the dissimilarity between the two distributions; this means that the network will optimize the weights by minimizing this dissimilarity, making the prediction distribution similar to the true one. Furthermore, we model the true label distribution as a Gaussian distribution, given that for the age estimation problem, errors have to be more reasonable possible (*e.g. consider a sample with true age 48: we do not only want the network to learn that the true age is 48 but also that saying 45 is a less critical error than saying 15*).

The second term (L_{er}, i.e. expectation regression loss) is the $\ell 1$ loss between the true label and the predicted age, that aims to further narrow the distance between the predicted age $\hat{y}_i$ and the ground truth label y_i.

$$L_{er} = |y_i - \hat{y_i}| \tag{4}$$

The predicted age $\hat{y_i}$ is computed with the *expectation regression* technique from the prediction distribution. In practice it is the statistic mean of the distribution:

$$\hat{y_i} = E[\hat{z_i^k}] = \sum_{k=1}^{100} k z_i^k \tag{5}$$

The λ parameter can be used to adjust the optimization objective of the network between distribution similarity and final prediction accuracy. For simplicity and since we dedicate more time to other network design choices, we use in all the experiments $\lambda = 1$ and $\sigma = 1$.

3.2 Classification stage

For classification stage we use a slightly modified version of L_r, which is identified as L_c (classification loss):

$$L_c = L_{ld} + (L_{er} - val\_MAE)^2 \tag{6}$$

The idea is to introduce in a smart way a kind of *MSE loss* for **narrowing down the standard deviation of different age groups**, using the obtained MAE on validation set from representation learning stage. Particularly, L_c is made of two terms: the first is the label distribution loss L_{ld}, which obviously has the same purpose already described above; the second is the square of the difference between L_{er} and *val_MAE*. Given that L_{er} is the $\ell 1$ loss between the true label and the predicted age, and *val_MAE* is the mean absolute error over the validation set computed on the model returned by representation learning stage, then the term $(L_{er} - val\_MAE)^2$ recalls the error standard deviation. Given that during classification stage the training is conducted with class-balanced sampling, minimizing this second term actually corresponds to minimize the standard deviation of different age groups. In other words, minimizing $(L_{er} - val\_MAE)^2$ corresponds to bring each MAE^j close to the global MAE value, which means that, during the classification stage, the network is trained with the objective of minimizing the error standard deviation among age groups.

To be more precise, *val_MAE* is defined in the following way:

$$val\_MAE = \sum_{i=1}^{M} \frac{|y_i - \hat{y_i}|}{M} \tag{7}$$

where M is the total number of validating images.

4 Experimental results

This section provides a presentation of the evaluation protocol, a detailed analysis of the conducted experiments, an interpretation of their results, an insight in terms of their repeatability and stability, and a prediction of our best model performance on the test set.

4.1 Evaluation protocol

Our proposed method is going to be evaluated in terms of accuracy and regularity. In particular, the contest organizers use a modified version of the Mean Absolute Error (MAE) for evaluating the accuracy of the submitted methods.

Suppose that the test set consists of K samples. The age prediction of a method for the i-th sample of the test set is p_i , while the real age label is r_i. Therefore, the absolute age estimation error on the i-th sample is:

$$e_i = |p_i - r_i| \tag{8}$$

The MAE is the average age estimation error over all the K samples of the test set:

$$MAE = \frac{\sum_{i=1}^{K} e_i}{K} \tag{9}$$

The following versions of MAE are computed:

1. MAE^1 is computed over the samples whose age annotation is in the range 1-10
2. MAE^2 is computed over the samples whose age annotation is in the range 11-20
3. MAE^3 is computed over the samples whose age annotation is in the range 21-30
4. MAE^4 is computed over the samples whose age annotation is in the range 31-40
5. MAE^5 is computed over the samples whose age annotation is in the range 41-50
6. MAE^6 is computed over the samples whose age annotation is in the range 51-60
7. MAE^7 is computed over the samples whose age annotation is in the range 61-70
8. MAE^8 is computed over the samples whose age annotation is in the range 70+

The $mMAE$, measure of accuracy that takes more into account the regularity of the error, is computed as follows:

$$mMAE = \frac{\sum_{j=1}^{8} MAE^j}{8} \tag{10}$$

The lower is the $mMAE$ achieved by a method, the higher is its average accuracy over the age groups. The standard deviation σ is the performance index used for evaluating the regularity of the methods over the different age groups:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{8} (MAE^j - MAE)^2}{8}} \tag{11}$$

The lower is the standard deviation achieved by a method, the higher is its regularity.
The final score is the Age Accuracy and Regularity (AAR) index, computed as follows:

$$AAR = \max(0; 5 - mMAE) + \max(0; 5 - \sigma) \tag{12}$$

The AAR index can assume values between 0 and 10. Methods which achieve $mMAE \geq 5$ and $\sigma \geq 5$ will obtain $AAR = 0$. A perfect method ($\sigma = 0$ and $mMAE = 0$) will achieve $AAR = 10$. Methods which achieve intermediate values of $mMAE$ and σ obtain intermediate values of AAR.

4.2 Experimental framework

Mivia Age Dataset has been randomly shuffled and then divided into 5 disjoint subsets of similar dimensions, so that each of them contains about 20% of all the provided images. The experiments described in this subsection, whose results are depicted 4.3, are conducted adopting the first 4 subsets as training set and the last subset as validation set. This split allows us to use the 80% of Mivia Age Dataset as training set and the remaining 20% as validation, which is a typical choice in machine learning field when splitting a given dataset.

For evaluating performance during experiments, we use exactly the same metrics described in 4.1. Since these same metrics are going to be used to judge our proposed model, this is the most reasonable choice. Moreover, in order to better understand our tested models behaviors, it is fundamental to take in consideration not only the AAR but also the single terms that contribute to it.
We conduct several experiments and report below the most significant ones, namely those that lead us to our final model.

- **Backbone Test.** This experiment consists of testing three different backbones for our problem. In particular, the networks taken in consideration are SeNet-50, VGG-16 and ResNet-50, all three pre-trained for face recognition on VGGFace2 Dataset [4]. All three are trained adopting exactly the same training procedure (that is described in detail in 2.2.3) and obviously the same training-validation split, as already said above. Then, the results are very meaningful: the differences in terms of performance are only due to the particular adopted backbone, so the outcomes are indicative to determine which of them performs better for our specific problem.
- **Classifier Test.** This experiment intends to investigate whether a different (in particular, deeper) classifier network could be useful to achieve better results than the ones granted by a single layer classifier. Particularly, all the three backbones are tested using a classifier made of three dense layers of 100 neurons each, since it seems to be a good compromise between an excessively elementary classifier and a too complex one.

4.3 Results

This subsection shows our most significant experiments results, with comments and interpretations, which guide us in the choice of our submitted model. Specifically, we provide for each experiment a detailed table, that includes the used backbone network, the obtained model, the number of training epochs and all the computed metrics on validation set: MAE, $mMAE$, σ, AAR, MAE^1, MAE^2, MAE^3, MAE^4, MAE^5, MAE^6, MAE^7, MAE^8.

4.3.1 Backbone Test

Table 1 shows the results (performance on validation set) of *Backbone Test*. It is useful to note that *R* and *C(100)* are respectively the notations with which, given a backbone, the models obtained at the end of representation learning (repr.) stage and classification (class.) stage are distinguished.

Table 1: Backbone Test. Performance on Validation Set.

Backbone	**Model**	**Epochs**	**MAE**	**mMAE**	σ	**AAR**	**MAE1**	**MAE2**	**MAE3**	**MAE4**	**MAE5**	**MAE6**	**MAE7**	**MAE8**
SeNet-50	R	26	2,04	2,59	1,20	6,21	5,23	2,17	1,72	2,10	2,25	1,90	2,27	3,06
	C(100)	12	2,27	2,43	0,36	7,22	3,18	2,59	2,16	2,17	2,42	2,21	2,28	2,41
VGG-16	R	23	1,94	2,62	1,44	5,95	5,81	2,32	1,62	1,97	2,09	1,85	2,24	3,02
	C(100)	15	2,14	2,29	0,37	7,33	3,15	2,39	1,99	2,19	2,21	2,11	2,12	2,20
ResNet-50	R	22	**1,67**	2,13	0,94	6,93	4,16	**1,88**	**1,39**	**1,73**	**1,81**	**1,64**	**1,92**	2,53
	C(100)	10	1,87	**2,01**	**0,32**	**7,67**	**2,72**	2,09	1,76	1,91	1,95	1,76	**1,92**	**1,98**

The following considerations can be done:

- The adopted two-stage training procedure is effective for all the three considered backbones, given that there is an average increase of about 1 in AAR from representation learning stage to classification. The main reason of this is the application of (6), that, as stated in subsection 3.2, is crucial for decreasing standard deviation error σ. In particular, such decrease is of about -0,84 on average from the first stage to the second one. In fact, it can be observed that the lowest σ decrease is associated to ResNet-50 and is -0,62, while the greatest σ decrease is given by VGG-16 and is -1,07. In general, σ reduction is always mostly attributable to a significant reduction in MAE^1 and MAE^8.

- The greatest AAR increase from repr. stage to class. stage is associated to VGG-16 and is 1,38; the lowest AAR increase is associated to ResNet-50 and is 0,74. SeNet-50, instead, shows an increase of 1,01.

- From repr. stage to class. stage MAE increases more less of the same quantity (0.2) for all the three backbones. Also $mMAE$ slightly increases from the first to the second stage, but with less regularity among backbones.

- In the light of the previous considerations, it can be noted for all the backbones that the improvement in AAR from repr. stage to class. stage is attributable only to a significant σ decrease, since $mMAE$ does not decrease, but actually slightly increases.

- The backbone that benefits the most from the decoupling learning strategy is VGG-16, while the one that benefits the least is ResNet-50. Despite of this, ResNet-50 is the backbone that performs the best, since at the end of repr. stage it already shows a relevant AAR of 6,93 which is clearly greater than the one reached by SeNet-50 (6,21) and VGG-16 (5,95).

- *ResNet-50 C(100)* is the model that achieves the best $mMAE$, σ, AAR, MAE^1, MAE^8. *ResNet-50 R* is the model that achieves the best MAE and all the best intermediate MAE^j (from MAE^2 to MAE^7, both included). All these best values are reported in **bold** in the table above. Particularly interesting is the best AAR: 7,67.

- *ResNet-50 C(100)* is the best in terms of AAR (7,67) thanks to a very low σ and a low $mMAE$, that are mostly the fruit of a great performance on the outermost age groups. This model, in fact, shows values for MAE^1 and MAE^8 that are significantly lower than the ones achieved by the others.

- In terms of AAR, the best is *ResNet-50 C(100)* with the already mentioned 7,67, then we find *VGG-16 C(100)* with 7,33 and at the end *SeNet-50 C(100)* with 7,22.

- The number of repr. stage epochs is higher than the class. stage one. This is due to a smaller patience during class. stage and mostly because the classifier is obviously much less complex than the backbone.

4.3.2 Classifier Test

Table 2 shows the results (performance on validation set) of *Classifier Test*. It is useful to note that *C(100,100,100)* is the notation that represents the adoption of a classifier network made of 3 dense layers of 100 neurons each.

Table 2: Classifier Test. Performance on Validation Set

Backbone	Model	Epochs	MAE	mMAE	σ	AAR	MAE1	MAE2	MAE3	MAE4	MAE5	MAE6	MAE7	MAE8
SeNet-50	C(100,100,100)	14	2,28	2,42	0,31	7,27	2,98	2,72	2,01	2,35	2,38	2,18	2,31	2,38
VGG-16	C(100,100,100)	11	2,24	2,56	0,75	6,69	4,32	2,40	2,08	2,26	2,24	2,31	2,51	2,34
Resnet-50	C(100,100,100)	10	**1,91**	**2,01**	**0,23**	**7,76**	**2,45**	**2,22**	**1,78**	**1,91**	**1,99**	**1,89**	**2,01**	**1,83**

The following considerations can be done:

- For SeNet-50 and ResNet-50, utilizing a deeper classifier improves performance on AAR respectively of 0.05 and 0.09: from 7,22 of *SeNet-50 C(100)* to 7,27 of *SeNet-50 C(100,100,100)* and from 7,67 of *ResNet-50 C(100)* to 7,76 of *ResNet-50 C(100,100,100)*. VGG-16, instead, shows a clear decrease in AAR: from 7,33 of *VGG-16 C(100)* to 6,69 of *VGG-16 C(100,100,100)*. These results are very interesting because, as already said at the end of Subsection 2.1, they show that when decoupling the learning procedure into representation learning and classification, the best classifier (in terms of its depth) is not always the simplest one, but depends on the specific problem, the network, and the level of discrimination among classes learned at representation stage. Particularly, when the backbone is VGG-16, the fact that as the classifier goes deeper, the performance gets worse probably means that the backbone network is already enough to learn a discriminative representation [10], while for SeNet-50 and ResNet-50 this is not true, so a deeper classifier reveals to be useful to improve performance, because it allows to better discriminate among ages.

- To be more precise, the network that benefits the most from increasing the classifier complexity is the one based on ResNet-50. In fact, comparing *ResNet-50 C(100,100,100)* and *ResNet-50 C(100)*, we can see that the first shows as MAE^1 and MAE^8 respectively 2,45 and 1,83, that are much lower than the corresponding ones achieved by the second, that are 2,72 and 1,98. Moreover, the first shows central MAE^j that are greater than the ones of the second, except for MAE^4, which remains the same. This allows to reduce σ (from 0,32 of the second to 0,23 of the first), while actually preserving the same $mMAE$. This is the key that allows *ResNet-50 C(100,100,100)* to reach a greater AAR than the one achieved by *ResNet-50 C(100)*.

- Also the network based on SeNet-50 improves thanks to a deeper classifier, but in a less evident fashion than the one which adopts ResNet-50. The network based on VGG-16, instead, as the classifier goes deeper, gets worse because all its MAE^j increase, then the same thing happens to σ, $mMAE$, AAR.

In the light of the reasons listed above, ***ResNet-50 C(100,100,100)* is the best candidate for the submission**. We decide to test the robustness of this choice in terms of repeatability through a K-fold cross validation.

4.4 Repeatability and stability of the results

Cross validation is a widely used evaluation method to estimate the performance of a machine learning model in heterogeneous scenarios, trying to predict how it will behave with unseen data. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into, hence the procedure is often called *k-fold cross-validation*. It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic/pessimistic estimate of the model skill than other methods, such as a simple train/test split.

Considering that previous tests have been conducted using a 80-20 split of the dataset, we choose to adopt a 5-fold cross validation. This way we make sure that at each step of the k fold the 80% of dataset is used for the training set and the remaining 20% for the validation set.

Table 3: K-fold Cross Validation. Validation Set performance for each fold and average performance.

Fold	Model	Epochs	MAE	mMAE	σ	AAR	MAE1	MAE2	MAE3	MAE4	MAE5	MAE6	MAE7	MAE8
0	R	18	1,60	2,03	0,89	7,08	3,97	1,92	1,32	1,65	1,73	1,55	1,81	2,31
	C(100,100,100)	18	1,19	1,28	0,16	8,56	1,49	1,46	1,06	1,21	1,22	1,17	1,32	1,28
1	R	27	1,99	2,49	1,08	6,43	4,92	2,35	1,71	2,04	2,10	1,93	2,20	2,65
	C(100,100,100)	17	1,29	1,39	0,20	8,41	1,46	1,75	1,19	1,24	1,32	1,26	1,50	1,39
2	R	19	1,64	2,15	1,03	6,82	4,35	1,86	1,34	1,70	1,73	1,62	2,04	2,55
	C(100,100,100)	18	1,19	1,31	0,20	8,49	1,52	1,55	1,08	1,95	1,21	1,15	1,34	1,41
3	R	17	2,62	2,88	0,62	6,51	4,30	2,68	2,43	2,63	2,88	2,48	2,68	2,94
	C(100,100,100)	14	1,27	1,39	0,23	8,38	1,46	1,81	1,19	1,20	1,27	1,28	1,49	1,44
4	R	22	1,89	2,30	0,73	6,97	3,44	2,11	1,55	1,91	2,06	1,85	2,44	3,06
	C(100,100,100)	13	1,86	2,00	0,29	7,71	2,61	2,13	1,72	1,91	1,92	1,77	1,94	1,96
Repr. Stage Averages			1,95	2,37	0,87	6,76	4,20	2,18	1,67	1,99	2,10	1,89	2,23	2,70
Class. Stage Averages			1,36	1,47	0,22	**8,31**	1,71	1,74	1,25	1,50	1,39	1,33	1,52	1,50

The results, that are reported in Table 3, are even better than the expectations. In fact, on average the AAR obtained is 8.31. This testifies the goodness of our method in terms of repeatability and stability. The good result obtained with *ResNet-50 c(100,100,100)* while performing the *Classifier Test* is not the product of fate but of good design choices.

Given these results, which validates our model architecture, loss functions adopted and training procedure, a reasonable choice for submission is to re train the *ResNet-50* model using all the provided dataset. However, since our procedure is based on a classification loss which embeds the validation MAE computed at the end of the repr. stage and since our training schedule uses the validation AAR to make decisions about learning rate changes and early stopping, we go for simply choosing randomly one of the models trained with folds 0,1,2,3, which show a pretty regularity in terms of AAR. Probably, the fold 4 underperformed due to the particular dataset random shuffle performed for k-fold subdivision. Obviously, performing more times the k-fold cross validation on different data random shuffles would give a better estimate of the average performance of the model. We submit the model trained on fold 0.

4.5 Prediction of the results on the test

To predict the value of AAR that our model will achieve on the test set, we adopt a statistical method that consists of the following: each group member predicts an upper and a lower AAR value such that the test AAR will be between these extremes; then these values are averaged obtaining common upper and lower AARs values.

Table 4: Prediction of AAR extremes on test set

	Upper AAR	**Lower AAR**
Salvatore Grimaldi	7,4	8
Luigi Schiavone	7,5	7,9
Roberto Falcone	7,2	8
Andrea De Gruttola	6,8	7,2
Average	**7,2**	**7,7**

Considering these predictions, we expect that the evaluation of the submitted model on the test set will result in a AAR between 7,2 and 7,7. Of course, our predictions are mostly driven by the results depicted in Table 3 and the absence of significant overfitting.

5 Conclusions

The results of the submitted model are very good, especially when compared to those of *”Guess The Age 2021”* [7]. This means that the decoupling approach proves to be good for the problem of age estimation with an imbalanced dataset, especially when combined with a slightly deeper classifier; the use of specially designed loss functions, the DAE approach, and the adoption of a pre-trained backbone are crucial. The results, in fact, seem to reward our choices and our significant effort. It is worth to note that the proposed system, since it consists of a single network, is obviously faster, in general, than a ensemble system. Just for give an idea of the inference time, the submitted network takes about 0,05 s on average to predict an image, using a NVIDIA Tesla T4 GPU.

Given more time, an interesting idea would be to combine our design choices with the use of a GAN [11] to generate samples of the less represented age groups, with the aim of ensuring even more balanced performance.

References

[1] Grigory Antipov, Moez Baccouche, Sid-Ahmed Berrani, and Jean-Luc Dugelay. Apparent age estimation from face images combining general and children-specialized deep learning models. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 96–104, 2016.

[2] Grigory Antipov, Moez Baccouche, Sid-Ahmed Berrani, and Jean-Luc Dugelay. Effective training of convolutional neural networks for face-based gender and age prediction. *Pattern Recognition*, 72:15–26, 2017.

[3] Zenghao Bao, Zichang Tan, Yu Zhu, Jun Wan, Xibo Ma, Zhen Lei, and Guodong Guo. Lae: long-tailed age estimation. In *International Conference on Computer Analysis of Images and Patterns*, pages 308–316. Springer, 2021.

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[5] Vincenzo Carletti, Antonio Greco, Gennaro Percannella, and Mario Vento. Age from faces in the deep learning revolution. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2113–2132, 2019.

[6] Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2401–2412, 2013.

[7] Antonio Greco. Guess the age 2021: Age estimation from facial images with deep convolutional neural networks. In *International Conference on Computer Analysis of Images and Patterns*, pages 265–274. Springer, 2021.

[8] Antonio Greco, Alessia Saggese, Mario Vento, and Vincenzo Vigilante. Effective training of convolutional neural networks for age estimation based on knowledge distillation. *Neural Computing and Applications*, pages 1–16, 2021.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[11] Yu-Hong Lin, Chia-Hao Tang, Zhi-Ting Chen, Gee-Sern Jison Hsu, Md Shopon, and Marina Gavrilova. Age-style and alignment augmentation for facial age estimation. In *International Conference on Computer Analysis of Images and Patterns*, pages 297–307. Springer, 2021.

[12] Zichang Tan, Jun Wan, Zhen Lei, Ruicong Zhi, Guodong Guo, and Stan Z Li. Efficient group-n encoding and decoding for facial age estimation. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2610–2623, 2017.

[13] Imad Eddine Toubal, Linquan Lyu, Dan Lin, and Kannappan Palaniappan. Single view facial age estimation using deep learning with cascaded random forests. In *International Conference on Computer Analysis of Images and Patterns*, pages 285–296. Springer, 2021.