Project Review:

Age Estimation

Artificial Vision

TEAM

- Gargiulo Michele
- Marchesano Riccardo
- Sabini Pietro
- Valitutto Andrea

PROBLEM

Given the image of the face of a subject, estimate its age rounded to the closest integer number.

DIFFICULTIES

1. Signs of people's ages shown in multiple ways.
2. Different lighting conditions.
3. Both male and female subjects.

DATASET DESCRIPTION

- The dataset is made up of 3.3 million images of ~9.000 different subjects (identities). Each identity has a number of images corresponding to different ages of the subject
- The identity-age distribution is not uniform, meaning that there are more images of a certain age of each subject, than there are of other ages
- The age is represented by a float number
- Each identity has a different number of images for each age
- Not every identity has all the possible ages
- Age range is different among different identities

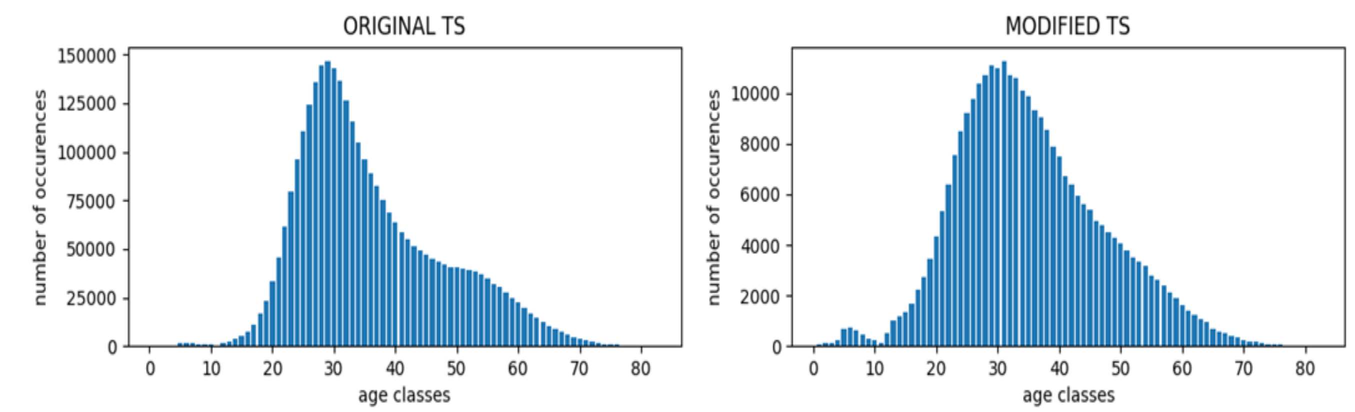
TRAINING/VALIDATION SET

- **Training set:** 255998 samples (80% of 319998)
- **Validation set:** 63999 samples (20% of 319998)

For each identity, the age range was divided into 4 groups and 9 images were taken for each group. The remaining 2 were taken on an experimental basis by the group that would have led to a distribution closer to the original one. Obviously, they were taken from the most common one, and not from the other bands, otherwise they would no longer have been available for a test set.

TRAINING/VALIDATION SET

- On the left : original distribution
- On the right: distribution after splitting



PROPOSED SOLUTION

ARCHITECTURE

- ResNet-50 using VGGFace2 weights
- ResNet-50 is an architecture particularly suited for facial-feature extraction, 50 layers deep
- 6 layers were added after the feature-extraction layers, reported in the next slide

ARCHITECTURE MODIFICATION

Layer	Description	# of parameters
Flatten	Necessary to flatten the tensor to 1 dimension. Size is 2048, and represents the feature vector	0
Dense (2048, relu)	First Dense Layer, using Relu activation. Same size of Flatten layer	4196352
Dropout (0.5)	First Dropout layer, used to reduce overfitting. Probability set to 50%	0
Dense (512, relu)	Second Dense layer, 512 is size, with Relu activation	1049088
Dropout (0.5)	Second Dropout layer, used to reduce overfitting. Probability set to 50%	0
Dense (101, Softmax)	Last dense layer. Size equal to the number of classes considered in the problem. Softmax activation allows the model to output a 101-sized one-hot encoded vector	51813

ARCHITECTURE MODIFICATION

activation_45 (Activation)	(None, 7, 7, 2048)	0	add_14[0][0]
conv5_3_1x1_reduce (Conv2D)	(None, 7, 7, 512)	1048576	activation_45[0][0]
conv5_3_1x1_reduce/bn (BatchNor	(None, 7, 7, 512)	2048	conv5_3_1x1_reduce[0][0]
activation_46 (Activation)	(None, 7, 7, 512)	0	conv5_3_1x1_reduce/bn[0][0]
conv5_3_3x3 (Conv2D)	(None, 7, 7, 512)	2359296	activation_46[0][0]
conv5_3_3x3/bn (BatchNormalizat	(None, 7, 7, 512)	2048	conv5_3_3x3[0][0]
activation_47 (Activation)	(None, 7, 7, 512)	0	conv5_3_3x3/bn[0][0]
conv5_3_1x1_increase (Conv2D)	(None, 7, 7, 2048)	1048576	activation_47[0][0]
conv5_3_1x1_increase/bn (BatchN	(None, 7, 7, 2048)	8192	conv5_3_1x1_increase[0][0]
add_15 (Add)	(None, 7, 7, 2048)	0	conv5_3_1x1_increase/bn[0][0] activation_45[0][0]
activation_48 (Activation)	(None, 7, 7, 2048)	0	add_15[0][0]
avg_pool (AveragePooling2D)	(None, 1, 1, 2048)	0	activation_48[0][0]
flatten (Flatten)	(None, 2048)	0	avg_pool[0][0]
dense (Dense)	(None, 2048)	4196352	flatten[0][0]
dropout (Dropout)	(None, 2048)	0	dense[0][0]
dense_1 (Dense)	(None, 512)	1049088	dropout[0][0]
dropout_1 (Dropout)	(None, 512)	0	dense_1[0][0]
Logits (Dense)	(None, 101)	51813	dropout_1[0][0]

Total params: 28,858,405
Trainable params: 28,805,285
Non-trainable params: 53,120

PRE-PROCESSING

- The main step implemented for pre-processing is the approach suggested by VGGFace2 authors, which is face normalization
- The applied technique is to subtract, for each image, the average of the 3 color channels
- The function used to perform such pre-processing was already coded and called ‘*mean_std_normalize*’, provided by the MiviaLab framework, in the ‘*dataset_tools*’ section

DATA AUGMENTATION

- The ‘*VGGFace2*’ mode has been specified for data augmentation.
- it performs random variations in terms of *flip*, *brightness*, *contrast* and *grayscale conversion*
- Other augmentation were tested, but they only resulted in worse performances or slightly better performances but with significantly prolonged training times
- A time-performance trade-off was considered for augmentation choice

LOSS FUNCTION

Ordinal Categorical Cross-entropy

- This is a Keras implementation of a loss function for ordinal datasets, based on the built-in categorical crossentropy loss.
- The assumption is that the relationship between any two consecutive categories is uniform, for example,

{[1, 0, 0, 0], [0, 0, 1, 0]}

- will be penalised to the same extent as

{[0, 1, 0, 0], [0, 0, 0, 1]}

- where {x, y} are the (truth, prediction) pairs.

METRIC

- Custom MAE (Mean Aboslute Error): is used to determine the model performance, but it is implemented in order to calculate the MAE based on the distance between the predicted classes and the real ones

- The weights whose model has the best custom MAE on validation are saved

TRAINING

- Mixed approach between Training from scratch and Fine tuning
- Warm-up phase: many models were tested, with different augmentations, final-layers architectures, weights and training parameters
- Each model Fine-tuned for 30 epochs
- Once the best model had been selected, a training phase for all layers was performed, resuming from the weights found before (for 50 epochs)

TRAINING PROCEDURES

Callback Lists :

- *Early-stopping*: was implemented to reduce the training times (monitor='val_loss', min_delta=0,002, patience=15)
- ***TensorBoard*: used to plot the traning and validation**
- ***ModelChackpoint***: Used to save the best model on 'val_mae'
- ***Reduction of lr***: It is used '*ReduceLROnPleatau*' from keras to reduce lr when the monitor not improve. (monitor='val_loss', factor=0.2, patience=5, min_lr=0.001).

RESULTS

The following report shows different results for the various models tested. The best model is Resnet50 at 41° epoch

Model	Weights	Training MAE	Validation MAE
Vgg16	Imagenet	1.9	4.5
MobileNet	Imagenet	2.3	4.8
Senet	imagenet	2.2	3.7
ResNet50	VggFace2	1,30	2.51

THANKS FOR YOUR ATTENTION