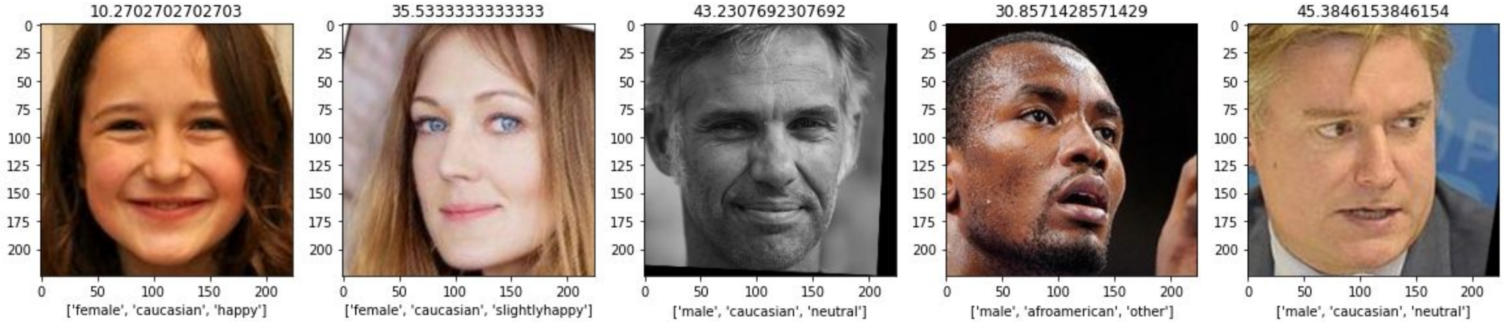
Practical Sessions Detailed

Dr. Julio C. S. Jacques Junior

julio.silveira@ub.edu

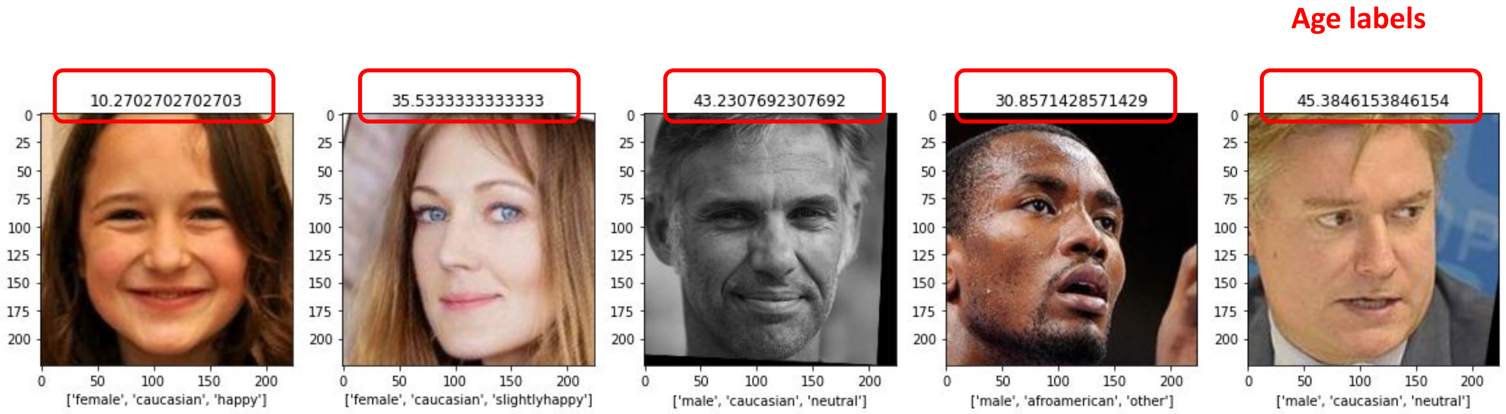
Problem: Automatic Age Perception

- You will need to solve a **regression** problem
- Given a face image, regress the **perceived age**



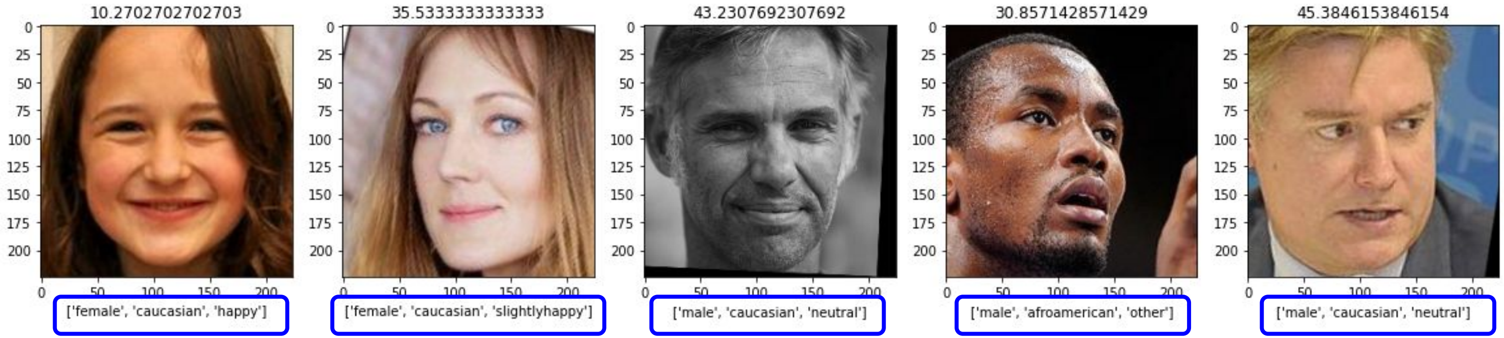
Problem: Automatic Age Perception

- You will need to solve a **regression** problem
- Given a face image, regress the **perceived age**



Problem: Automatic Age Perception

- You will need to solve a **regression** problem
- Given a face image, regress the **perceived age**

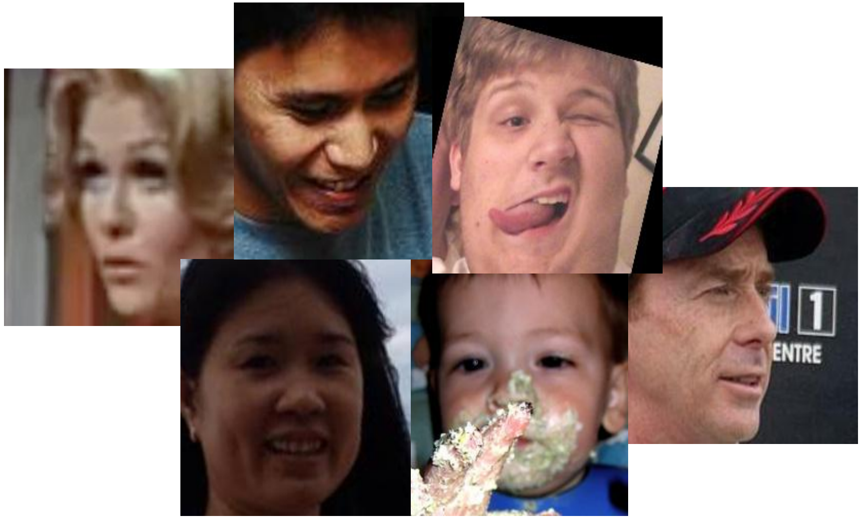


Metadata

Gender (male / female)
Ethnicity (asian / afroamerican / caucasian)
Facial expression (neutral / slightly-happy / happy / other)

Problem: Automatic Age Perception

- **It looks simple but** several challenges are involved
    - Pose variation
    - Different image qualities
    - Different illumination conditions
    - Occlusions, etc



Dataset: Appa-Real Age Dataset

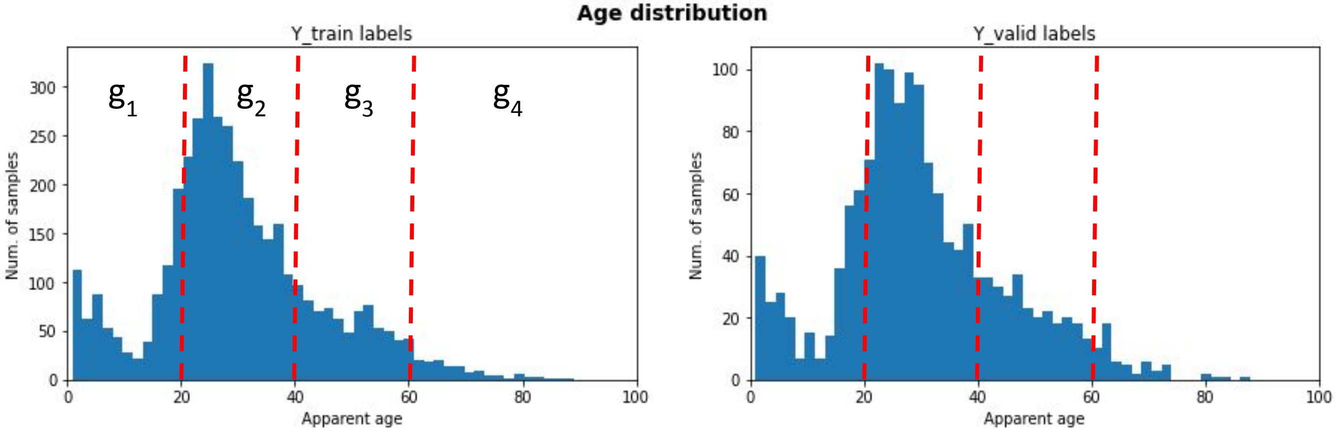
- The data is divided in:
    - **Train** (4065 images),
    - **Validation** (1482 images) and
    - **Test** (1978 images) set (*without labels*)

- Matadata is also provided:
    - **Gender**: male / female
    - **Ethnicity**: asian / afroamerican / caucasian
    - **Facial expression**: neutral / slightly-happy / happy / other

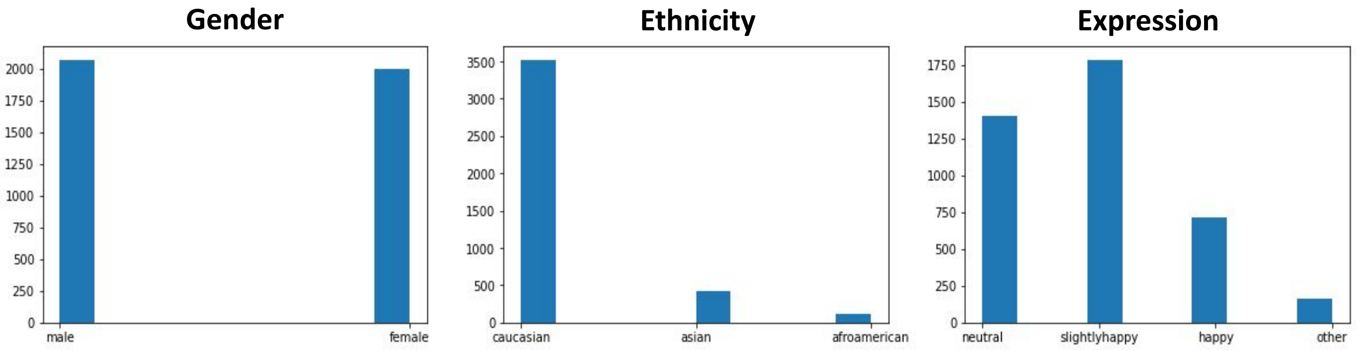
http://chalearnlap.cvc.uab.es/challenge/13/track/13/description/

- **Dataset is biased** *w.r.t* different attributes

Training data distribution: Age



Training data distribution: Metadata



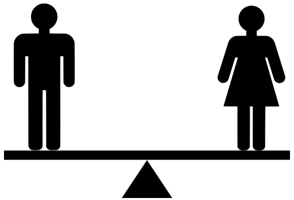
Your Goal: maximize accuracy & minimize the bias scores

- Reduced *Mean Absolute Error (MAE)* → *Global*

- Reduced *Bias scores*
  - **Gender** bias (2 groups)
  - **Age** bias (4 age groups)
  - **Ethnicity** bias (3 groups)
  - **Facial expression** bias (4 groups)

➔ **Bias metric goal:** to minimize the MAE (E) difference among different sub-groups (N), given an attribute (A).

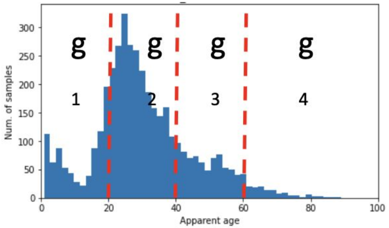
$$B_A = \frac{1}{(N^2 - N)/2} \sum_{i=1}^{N} \sum_{j=1}^{N} |E_i - E_j|, \forall i, j \in \mathbb{N}^*, \text{if } i < j$$



Ideally, the method should predict with similar accuracy for all different subgroups

Bias metric (goal)

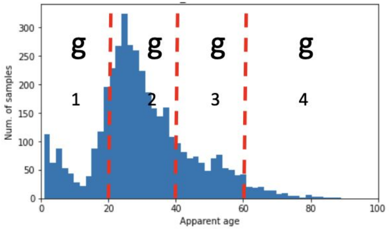
- For example, let’s consider the **Age attribute**, where we have 4 sub-groups (N=4).



    - First, we compute an error measure (E_n) for each sub-group.
    - Then, we compute the absolute difference among the N sub-groups.
    - The final bias score is the mean value of these diffencens.

Bias metric (goal)

- For example, let’s consider the **Age attribute**, where we have 4 sub-groups (N=4).



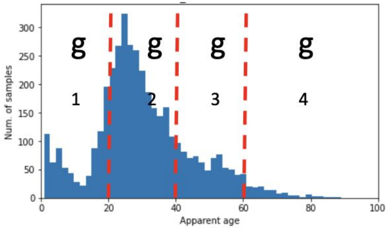
  - First, we compute an error measure (E_n) for each sub-group.

E_1, E_2, E_3 and E_4

  - Then, we compute the absolute difference among the N sub-groups.
  - The final bias score is the mean value of these diffencens.

Bias metric (goal)

- For example, let's consider the **Age attribute**, where we have 4 sub-groups (N=4).



  - First, we compute an error measure (E_n) for each sub-group.

E_1, E_2, E_3 and E_4

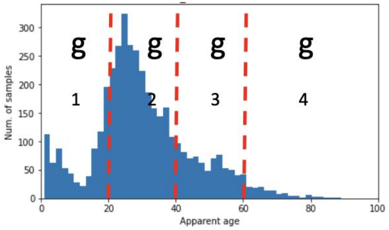
  - Then, we compute the absolute difference among the N sub-groups.

$$\begin{aligned}
D_{2,1} &= |E_1 - E_2| \\
D_{3,1} &= |E_1 - E_3| \\
D_{4,1} &= |E_1 - E_4| \\
D_{3,2} &= |E_2 - E_3| \\
D_{4,2} &= |E_2 - E_4| \\
D_{4,3} &= |E_3 - E_4|
\end{aligned}$$

  - The final bias score is the mean value of these diffencens.

Bias metric (goal)

- For example, let's consider the **Age attribute**, where we have 4 sub-groups (N=4).



  - First, we compute an error measure (E_n) for each sub-group.

E_1, E_2, E_3 and E_4

  - Then, we compute the absolute difference among the N sub-groups.

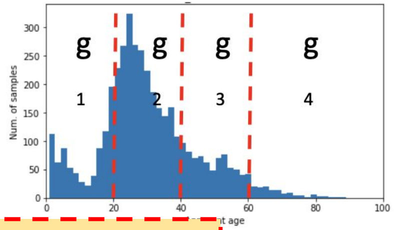
$$\begin{aligned}
D_{2,1} &= |E_1 - E_2| \\
D_{3,1} &= |E_1 - E_3| \\
D_{4,1} &= |E_1 - E_4| \\
D_{3,2} &= |E_2 - E_3| \\
D_{4,2} &= |E_2 - E_4| \\
D_{4,3} &= |E_3 - E_4|
\end{aligned}$$

  - The final bias score is the mean value of these diffencens.

$$B_A = \frac{(D_{2,1}+D_{3,1}+D_{4,1}+D_{3,2}+D_{4,2}+D_{4,3})}{6}$$

Bias metric (goal)

- For example, let's consider the **Age attribute**, where we have 4 sub-groups (N=4).



  - First, we co ... d E_4
    each sub-g
  - Then, we c ... $E_2|$ $E_3|$ $E_4|$ $E_3|$ $E_4|$ $E_4|$
    among the
  - The final bi
    diffencens.

This will "force" us to minimize the error difference among these sub-groups.

In other words, **making them having similar accuracy**.

$$B_A = \frac{(D_{2,1}+D_{3,1}+D_{4,1}+D_{3,2}+D_{4,2}+D_{4,3})}{6}$$

The dynamics of the practical sessions

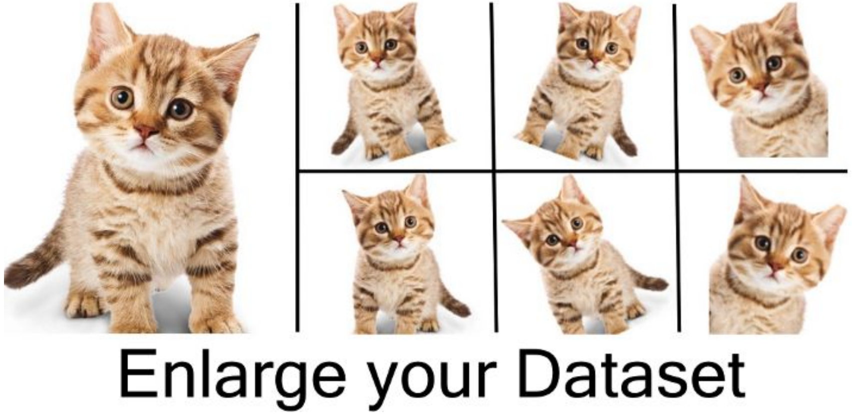
Tasks
Deliverables
Final Project Presentation

Task 1: Intelligent data augmentation

Task 1

- **Challenge:** deal with multiple attributes together (age, gender, ethnicity, expression)
- Ex.: generating new data for a particular group may affect the data distribution of other groups or subgroups

Compared with baseline results…



source: https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

What you should **avoid**

- Making **minor modifications of the starting kit**:

```
# from
x_blur = cv2.GaussianBlur(x,(5,5),1.0)

# to
x_blur = cv2.GaussianBlur(x,(7,7),1.0)
```

- Augment the data for **the same attribute only**:

```
if Y_train[i]*100>=60:
  # flip
  X_train_augmented.append(cv2.flip(X_train[i], 1))
  Y_train_augmented.append(Y_train[i])
```

Be creative!

Task 2: Custom Loss (without data augmentation)

- **Challenge:** deal with multiple attributes together (age, gender, etc)
- Ex.: creating a "customized loss" which gives more weight to people having less samples in train data.
- During training, the model may better learn how to make predictions on those cases while trying to minimize the loss.
    - The starting-kit will consider the age range only, based on different age ranges. This way, we believe the model will be able to generalize a little bit better.

Compared with baseline results…

Compared with task 1…

What you should avoid

- Use **the same strategy** as the starting kit:

$$w_j = n_{samples} / (n_{classes} * n_{samples,j}),$$

- Consider **the same attribute only**.

```
for i in range(0,Y_train.shape[0]):
    if(Y_train[i]*100<20):
      sample_weights.append(w[0])
    if(Y_train[i]*100>=20 and Y_train[i]*100<40):
      sample_weights.append(w[1])
    if(Y_train[i]*100>=40 and Y_train[i]*100<60):
      sample_weights.append(w[2])
    if(Y_train[i]*100>=60):
      sample_weights.append(w[3])
```

Be creative!

Optional Task 3: Surprise us

- **Exploit your creativity as much as you can ("surprise us")**
- **Challenge:** deal with multiple attributes together (age, gender, etc)
  - With/without data augmentation and/or custom loss.

Source: https://www.pexels.com

Working in **groups**

- Our proposal is to work in **groups** → check the details on **Virtual Campus**.
    - Stimulate collaborative work
    - Receive quick feedback

- **Groups should be defined ASAP, as the Tasks (and deliverables) will be defined by the and of this class;**
    - Please, include the information about your group in the **shared doc** available on **Virtual Campus**.

Workflow

- **Colab** “Starting kit”
- Make predictions
- **Submission file** (.zip)

Submitting your predictions

Receive Real-time Feedback

Colab File + Report:
- Deliverables (Task 1, 2 & the optional Task)

Results

#	User	Entries	Date of Last Entry	Ranking ▲	Gender (bias) ▲	Expression (bias) ▲	Ethnicity (bias) ▲	Age (bias) ▲	MAE ▲
1	[illegible]	1	02/01/21	1.200000	0.293883 (1)	1.027783 (2)	2.130759 (1)	7.460989 (1)	10.975767 (1)
2	[illegible]	1	02/01/21	1.800000	0.628002 (2)	0.819692 (1)	2.447512 (2)	8.988890 (2)	11.141265 (2)

analysed **biases**

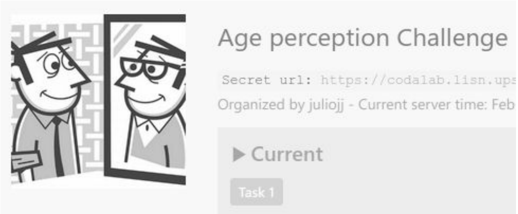
“**Accuracy**”

Ranking: *average rank position*

Colab & Starting-kit

- **Colab** “Starting kit”
- Make predictions
- **Submission file** (.zip)

CodaLab



Submitting your predictions

Receive Real-time Feedback

Colab File + Report:
- Deliverables (Task 1, 2 & the optional Task)

Results

#	User	Entries	Date of Last Entry	Ranking ▲	Gender (bias) ▲	Expression (bias) ▲	Ethnicity (bias) ▲	Age (bias) ▲	MAE ▲
1	[illegible]	1	02/01/21	1.200000	0.293883 (1)	1.027783 (2)	2.130759 (1)	7.460989 (1)	10.975767 (1)
2	[illegible]	1	02/01/21	1.800000	0.628002 (2)	0.819692 (1)	2.447512 (2)	8.988890 (2)	11.141265 (2)

analysed **biases**

“**Accuracy**”

Ranking: *average rank position*

Colab & Starting-kit

- Allow you to use CPU/GPU units on the cloud (GPU: *not unlimited*)
- We have prepared a **jupyter notebook** where you can:
  - Get introduced to the problem **progressively**
  - **Download** the data (train/valid/test)
  - **Visualize** the data/metadata
  - Run **baseline** methods (*code available*)
  - **Train / Load** pre-trained models
- Edit / **adapt / improve** the baseline methods

```
import h5py
import tensorflow as tf
from tensorflow.keras.models import Model, load_mode
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam

# loading the pretrained model
model = tf.keras.models.load_model('./model/weights
print(model.summary())
```

```
activation_43 (Activation)       (None, 7, 7, 2048)
conv5_2_1x1_reduce (Conv2D)      (None, 7, 7, 512)
conv5_2_1x1_reduce/bn (BatchNor  (None, 7, 7, 512)
activation_44 (Activation)       (None, 7, 7, 512)
conv5_2_3x3 (Conv2D)             (None, 7, 7, 512)
conv5_2_3x3/bn (BatchNormalizat  (None, 7, 7, 512)
activation_45 (Activation)       (None, 7, 7, 512)
conv5_2_1x1_increase (Conv2D)    (None, 7, 7, 2048)
```

Colab & Starting-kit

- Allow you to use CPU/GPU units on the cloud (GPU: *not unlimited*)
- We have prepared a **jupyter notebook** where you can:
  - Get introduced to the problem **progressively**
  - **Download** the data (train/valid/test)
  - **Visualize** the data/metadata
  - Run **baseline** methods (*code available*)
  - **Train / Load** pre-trained models
- Edit / **adapt / improve** the baseline methods

```
import h5py
import tensorflow as tf
from tensorflow.keras.models import Model, load_mode
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam

# loading the pretrained model
model = tf.keras.models.load_model('./model/weights
print(model.summary())
```

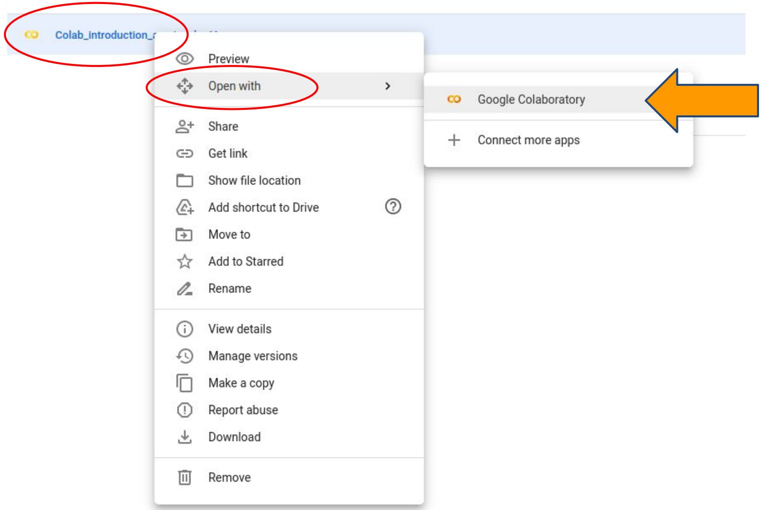
```
activation_43 (Activation)      (None, 7, 7, 2048)
conv5_2_1x1_reduce (Conv2D)     (None, 7, 7, 512)
conv5_2_1x1_reduce/bn (BatchNor (None, 7, 7, 512)
activation_44 (Activation)      (None, 7, 7, 512)
conv5_2_3x3 (Conv2D)            (None, 7, 7, 512)
conv5_2_3x3/bn (BatchNormalizat (None, 7, 7, 512)
activation_45 (Activation)      (None, 7, 7, 512)
conv5_2_1x1_increase (Conv2D)   (None, 7, 7, 2048)
```

The task is already solved!!

We expect you to go beyond the starting-kit!

Colab & Starting-kit: “**Hello Colab**”

- Upload the provided “**.ipynb**” file to your Drive
- Open the file with “**Google Collaboratory**” colab



Colab & Starting-kit: “**Hello Colab**”

- Data loading
- Visualization
- Modeling
- Training (*stop & continue*)
- Evaluation

Recommendation:

A) Press **“Play”** and get used with everything

B) Edit → Improve

```
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

# load a model and train history (defined and trained
# as below, trained for 38 epochs)
#--------------------------
LOAD_BEST_MODEL_ST1 = True # (training only the last FC layers)
#--------------------------


if(LOAD_BEST_MODEL_ST1==True):
  # downloading the trained model
  !wget https://www.dropbox.com/s/x51d08o20ybzqto/best_model_st1.zip
  # decompressing the data
  with ZipFile('best_model_st1.zip','r') as zip:
    zip.extractall()
    print('Model decompressed successfully')
  # removing the .zip file after extraction  to clean space
  !rm best_model_st1.zip

else:
  # defining the early stop criteria
  es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=
  # saving the best model based on val_loss
  mc = ModelCheckpoint('/content/gdrive/MyDrive/temp/best_model.h5', moni

  # defining the optimizer
  model.compile(tf.keras.optimizers.Adam(learning_rate=1e-5),loss=tf.kera

  # training the model
  history = model.fit(X_train, Y_train, validation_data=(X_valid, Y_valid
```

Codalab Competition

- **Colab** "Starting kit"
- Make predictions
- **Submission file** (.zip)

CodaLab

Submitting your predictions

Receive Real-time Feedback

Colab File + Report:
- Deliverables (Task 1, 2 & the optional Task)

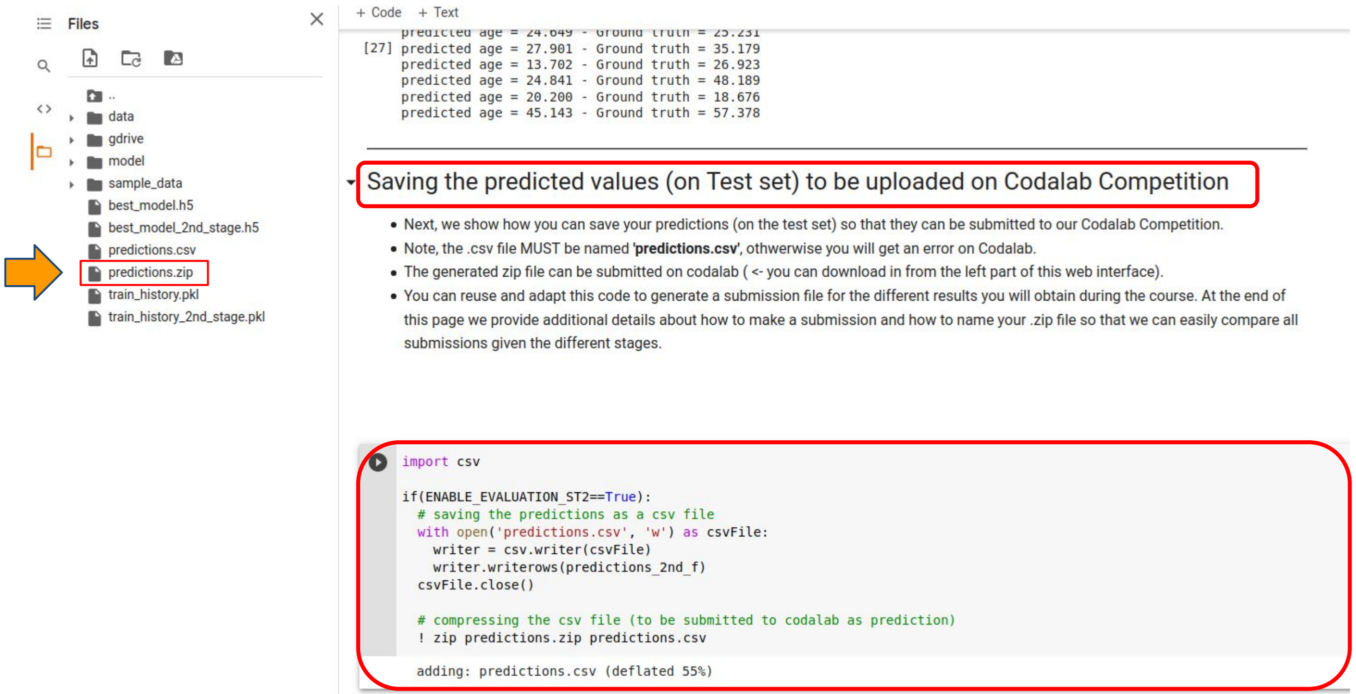
#	User	Entries	Date of Last Entry	Ranking ▲	Gender (bias) ▲	Expression (bias) ▲	Ethnicity (bias) ▲	Age (bias) ▲	MAE ▲
					Results				
1	[illegible]	1	02/01/21	1.200000	0.293883 (1)	1.027783 (2)	2.130759 (1)	7.460989 (1)	10.975767 (1)
2	[illegible]	1	02/01/21	1.800000	0.628002 (2)	0.819692 (1)	2.447512 (2)	8.988890 (2)	11.141265 (2)

Ranking: *average rank position*

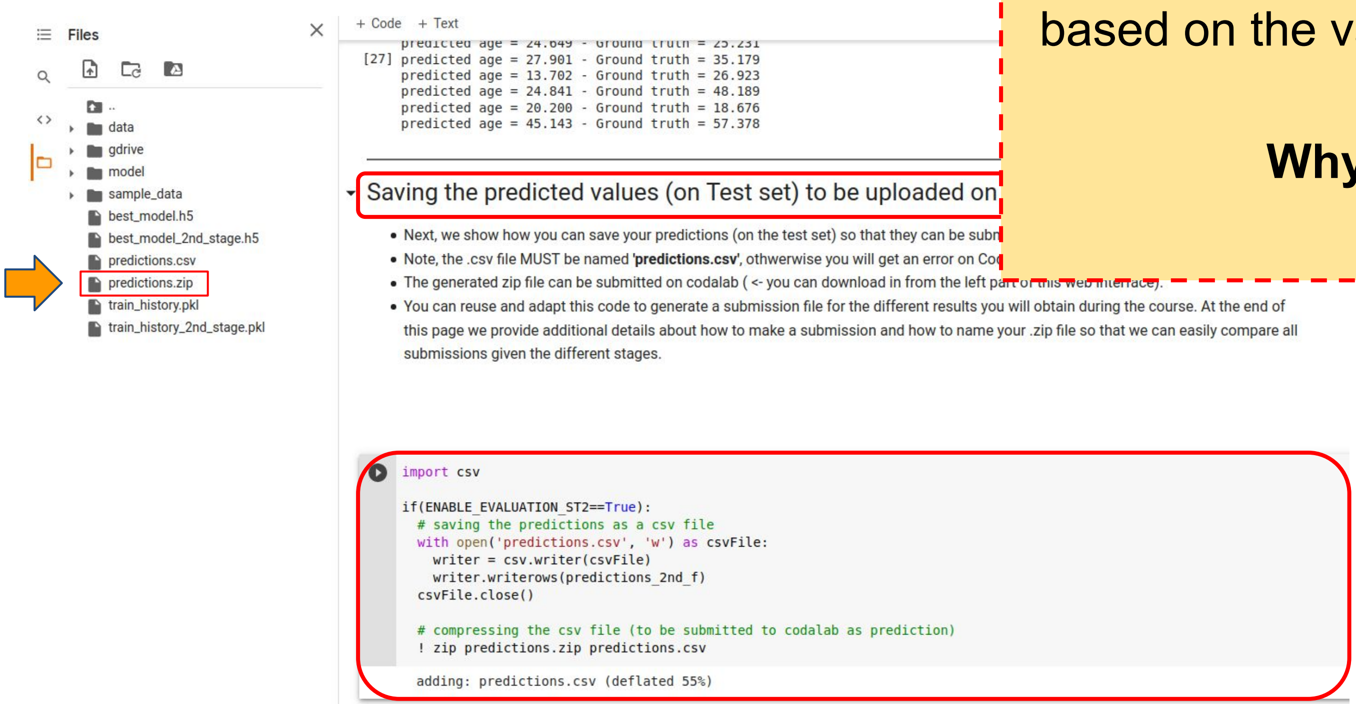
analysed **biases**

"**Accuracy**"

Submission file: Colab → Codalab



Submission file: Colab → Codalab



After **model selection** and hyperparameter tuning, based on the validation set!

Why?

Codalab Competition: **main goal → motivation**

1. **Motivate you** to improve your method and results
    a. Compared to your previous submissions
    b. Compared to your colleagues and other students
2. Simulate a real scenario in research (to **motivate you**)
3. Have fun while learning new skills (to **motivate you**)

#	User	Entries	Date of Last Entry	Ranking ▲
1	[illegible]	1	02/01/21	1.200000
2	[illegible]	1	02/01/21	1.800000

IMPORTANT: The Rank position on Codalab **WON'T be considered** for the evaluation!

Codalab Competition: **Submitting your results**

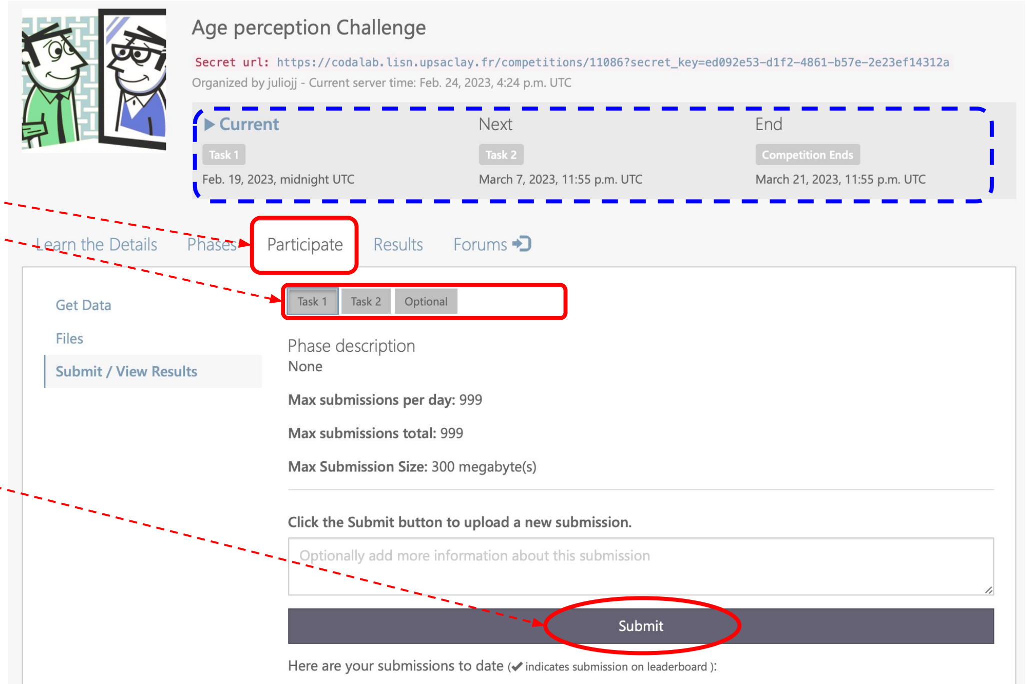
1. **Register** on Codalab: https://codalab.lisn.upsaclay.fr
2. Access our Challenge

   Challenge link available on eCampus

3. **Participate**

   (*Task is automatically defined based on the* schedule)

4. **Submit** your file



Codalab Competition: **Real-time feedback**

Results									
#	User	Entries	Date of Last Entry	Ranking	Gender (bias)	Expression (bias)	Ethnicity (bias)	Age (bias)	MAE
1	[illegible]	1	02/01/21	1.200000	0.293883 (1)	1.027783 (2)	2.130759 (1)	7.460989 (1)	10.975767 (1)
2	[illegible]	1	02/01/21	1.800000	0.628002 (2)	0.819692 (1)	2.447512 (2)	8.988890 (2)	11.141265 (2)

analysed **biases**

“**Accuracy**”

Ranking: *average rank position*

= (1 + 2 + 1 + 1 + 1)/5 = **1.2**

Codalab Competition: **baselines**

Google colab

- **Colab** “Starting kit”
- Make predictions
- **Submission file** (.zip)

CodaLab

- **Baseline results obtained from the starting-kit, but trained for more epochs**
  - **With data augmentation (task 1)**
  - **Custom loss (task 2)**

me

Submitting your predictions

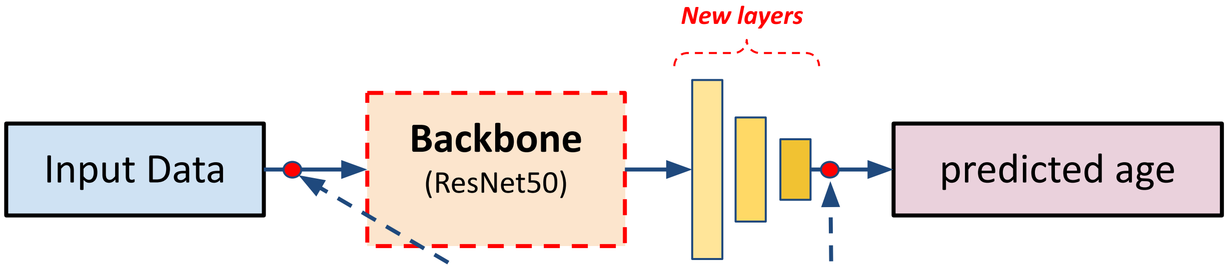
							bias)	Age (bias)	MAE
							(1)	7.460989 (1)	10.975767 (1)
2	[illegible]	1	02/01/21	1.800000	0.628002 (2)	0.819692 (1)	2.447512 (2)	8.988890 (2)	11.141265 (2)

Ranking: *average rank position*

analysed **biases**

“**Accuracy**”

Starting kit: **General Working Plan**



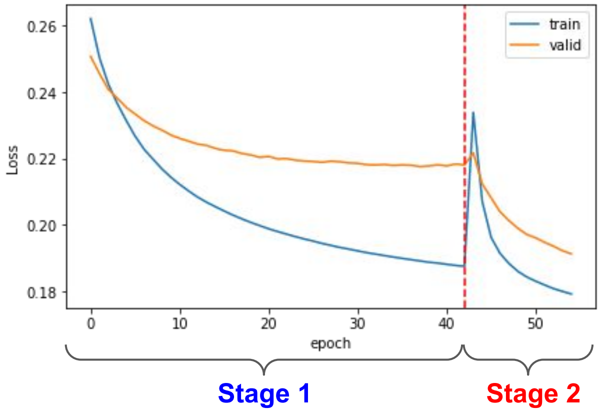
Task 1: play with hyperparameters, model definition, learning strategy, and **with data augmentation**.

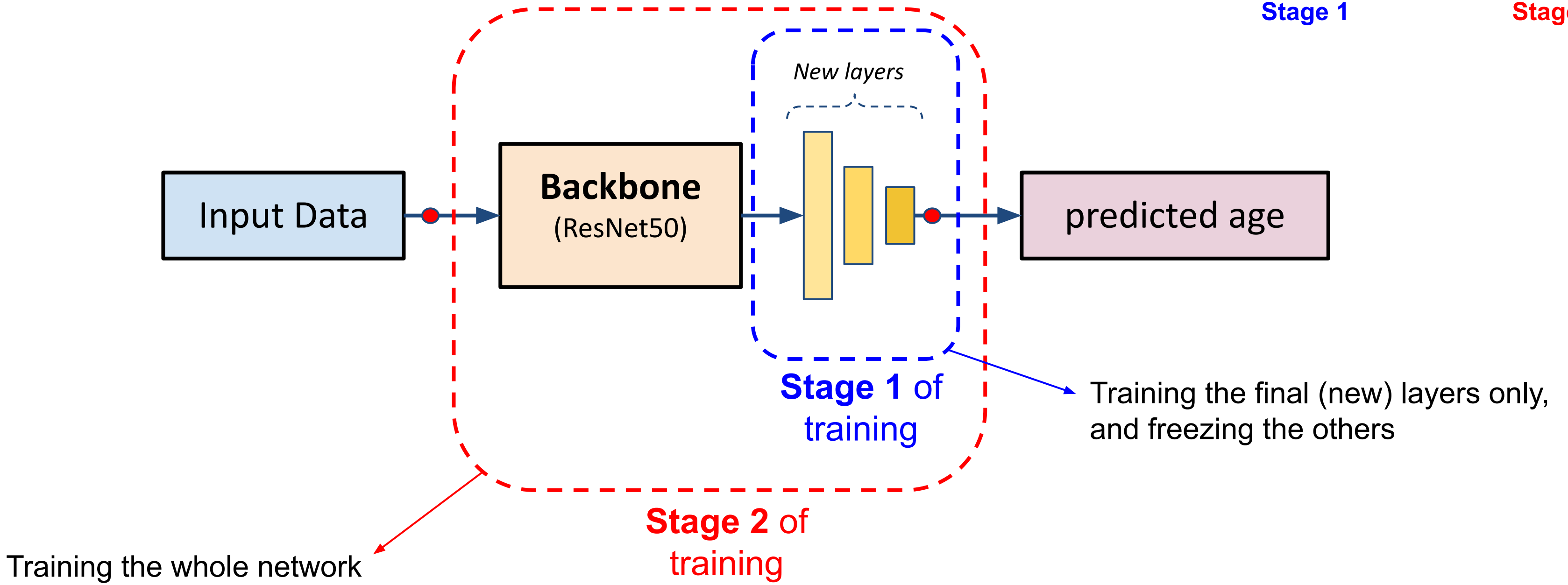
Task 2: fix the previous model and play with hyperparameters, learning strategy, custom loss / weighted samples (**without data augmentation**)

Optional (extra) Task: all together + "surprise us"

Starting kit: **Training Strategy**

- You are free to employ any training strategy you want





Suggestion for your experiments

- Try **different backbones** (https://keras.io/api/applications/)
- Use different **pretrained** models (or train from scratch)
- Play with different **hyperparameters**
  - Learning rate, num of epochs, batch size
- Play with different **regularizers**
  - Drop out, early stopping
- Play with different **training strategies**
  - One single stage
  - Fine-tuning different blocks of a pre-trained model
  - Using different optimizers
- Define your experiments clearly and compare. Ex.:
  - Model A vs model B
  - Model A using hyperparameter x=0.1, x=0.01, x=0.001
  - Model B with and without pre-training
  - Etc

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3
DenseNet201	80	77.3%	93.6%	20.2M	402	127.2	6.7
NASNetMobile	23	74.4%	91.9%	5.3M	389	27.0	6.7
NASNetLarge	343	82.5%	96.0%	88.9M	533	344.5	20.0
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9
EfficientNetB1	31	79.1%	94.4%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.1%	94.9%	9.2M	186	80.8	6.5
EfficientNetB3	48	81.6%	95.7%	12.3M	210	140.0	8.8
EfficientNetB4	75	82.9%	96.4%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.6%	96.7%	30.6M	312	579.2	25.3
EfficientNetB6	166	84.0%	96.8%	43.3M	360	958.1	40.4
EfficientNetB7	256	84.3%	97.0%	66.7M	438	1578.9	61.6
EfficientNetV2B0	29	78.7%	94.3%	7.2M	-	-	-
EfficientNetV2B1	34	79.8%	95.0%	8.2M	-	-	-
EfficientNetV2B2	42	80.5%	95.1%	10.2M	-	-	-
EfficientNetV2B3	59	82.0%	95.8%	14.5M	-	-	-
EfficientNetV2S	88	83.9%	96.7%	21.6M	-	-	-
EfficientNetV2M	220	85.3%	97.4%	54.4M	-	-	-
EfficientNetV2L	479	85.7%	97.5%	119.0M	-	-	-
ConvNeXtTiny	109.42	81.3%	-	28.6M	-	-	-
ConvNeXtSmall	192.29	82.3%	-	50.2M	-	-	-
ConvNeXtBase	338.58	85.3%	-	88.5M	-	-	-
ConvNeXtLarge	755.07	86.3%	-	197.7M	-	-	-
ConvNeXtXLarge	1310	86.7%	-	350.1M	-	-	-

Suggestion for your experiments

- Try **different backbones** (https://keras.io/api/applications/)
- Use different **pretrained** models (or train from scratch)
- Play with different **hyperparameters**
  - Learning rate, num of epochs, batch size
- Play with different **regularizers**
  - Drop out, early stopping
- Play with different **training strategies**
  - One single stage
  - Fine-tuning different blocks of a pre-trained model
  - Using different optimizers
- Define your experiments clearly and compare. Ex.:
  - Model A vs model B
  - Model A using hyperparameter x=0.1, x=0.01, x=0.001.
  - Model B with and without pre-training
  - Etc

Suggestion for your experiments

- Try **different backbones** (https://keras.io/api/applications/)
- Use different **pretrained** models (or train from scratch)
- Play with different **hyperparameters**
  - Learning rate, num of epochs, batch size
- Play with different **regularizers**
  - Drop out, early stopping
- Play with different **training strategies**
  - One single stage
  - Fine-tuning different blocks of a pre-trained model
  - Using different optimizers
- Define your experiments clearly and compare. Ex.:
  - Model A vs model B
  - Model A using hyperparameter x=0.1, x=0.01, x=0.001.
  - Model B with and without pre-training
  - Etc

Suggestion for your experiments

- Try **different backbones** (https://keras.io/api/applications/)
- Use different **pretrained** models (or train from scratch)
- Play with different **hyperparameters**
  - Learning rate, num of epochs, batch size
- Play with different **regularizers**
  - Drop out, early stopping
- Play with different **training strategies**
  - One single stage
  - Fine-tuning different blocks of a pre-trained model
  - Using different optimizers
- Define your experiments clearly and compare. Ex.:
  - Model A vs model B
  - Model A using hyperparameter x=0.1, x=0.01, x=0.001.
  - Model B with and without pre-training
  - Etc

Suggestion for your experiments

- Try **different backbones** (https://keras.io/api/applications/)
- Use different **pretrained** models (or train from scratch)
- Play with different **hyperparameters**
  - Learning rate, num of epochs, batch size
- Play with different **regularizers**
  - Drop out, early stopping
- Play with different **training strategies**
  - One single stage
  - Fine-tuning different blocks of a pre-trained model
  - Using different optimizers
- Define your experiments clearly and compare. Ex.:
  - Model A vs model B
  - Model A using hyperparameter x=0.1, x=0.01, x=0.001.
  - Model B with and without pre-training
  - Etc

Suggestion for your experiments

- Try **different backbones** (https://keras.io/api/applications/)
- Use different **pretrained** models (or train from scratch)
- Play with different **hyperparameters**
  - Learning rate, num of epochs, batch size
- Play with different **regularizers**
  - Drop out, early stopping
- Play with different **training strategies**
  - One single stage
  - Fine-tuning different blocks of a pre-trained model
  - Using different optimizers
- Define your experiments clearly and compare. Ex.:
  - Model A vs model B
  - Model A using hyperparameter x=0.1, x=0.01, x=0.001.
  - Model B with and without pre-training
  - Etc

Suggestion for your experiments

- Try **different backbones** (https://keras.io/api/applications/)
- Use different **pretrained** models (or train from scratch)
- Play with different **hyperparameters**
  - Learning rate, num of epochs, batch size
- Play with different **regularizers**
  - Drop out, early stopping
- Play with different **training strategies**
  - One single stage
  - Fine-tuning different blocks of a pre-trained model
  - Using different optimizers
- Define your experiments clearly and compare. Ex.:
  - Model A vs model B
  - Model A using hyperparameter x=0.1, x=0.01, x=0.001.
  - Model B with and without pre-training
  - Etc

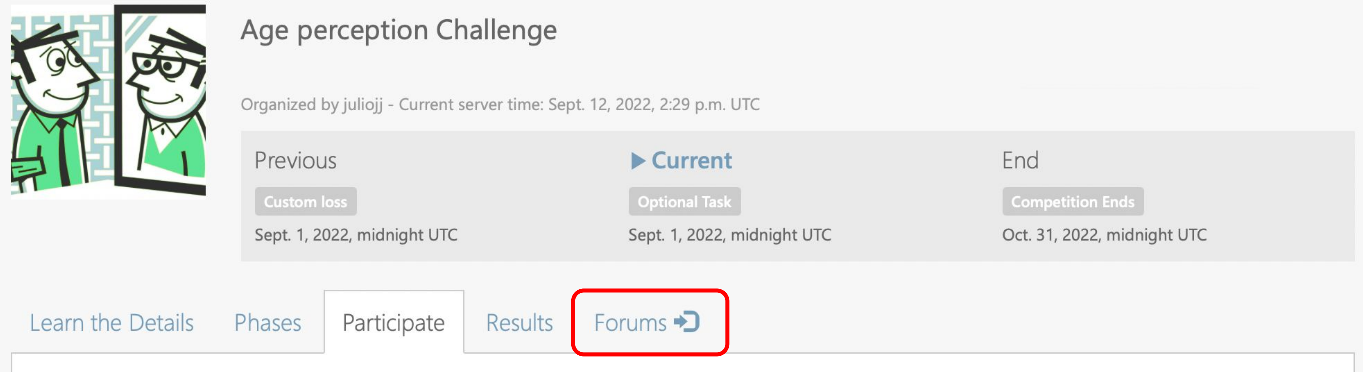
Remember:

The task is already solved!!

We expect you to go beyond the starting-kit!

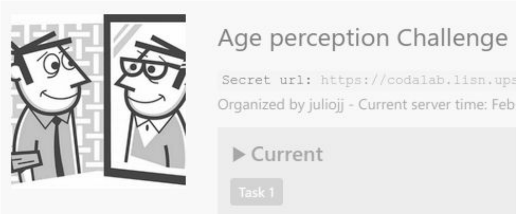
Forums

- You can use the Forum to **exchange experiences**, ask **questions** and report any **problem** (apart from the Virtual Campus).



Deliverables

- **Colab** "Starting kit"
- Make predictions
- **Submission file** (.zip)



Submitting your predictions

Receive Real-time Feedback

Colab File + Report:
- Deliverables (**Task 1, 2** & the optional Task)

Results

#	User	Entries	Date of Last Entry	Ranking	Gender (bias)	Expression (bias)	Ethnicity (bias)	Age (bias)	MAE
1	[illegible]	1	02/01/21	1.200000	0.293883 (1)	1.027783 (2)	2.130759 (1)	7.460989 (1)	10.975767 (1)
2	[illegible]	1	02/01/21	1.800000	0.628002 (2)	0.819692 (1)	2.447512 (2)	8.988890 (2)	11.141265 (2)

analysed **biases**

"**Accuracy**"

Ranking: *average rank position*

Report + Colab file

- For each deliverable: ***Task 1, 2*** and the ***Optional (extra) Task***
- **Your group** will have to deliver:
  - **Report document** (saved as **.pdf**) → **will receive more attention**
    - Detailing the proposed solution, with a strong analysis and discussion of the experiments and results.
  - ***Colab file*** (saved as **.ipynb**) → will be used to complement the report
    - Well documented and with clean code → please, remove basic information that is not **your contribution**.
- ***How to share these files?***
  - *Zip both files (**your_name_task_id.zip**) → submit it via **Virtual Campus** by the deadline (one .zip file per group to avoid inconsistencies)*

Report document Template

(Link for download on **Virtual Campus**)

HEADER

1. SUMMARY OF CONTRIBUTIONS

2. EXPERIMENTS AND RESULTS

3. FINAL REMARKS

Go direct to the point!

OPT 2 – COMPUTER VISION (2022)
REPORT: **Task 1&2** (or "optional Task 3")

Group members:
full name (1), <email_1@domain>, Codalab_user_1
full name (2), <email_2@domain>, Codalab_user_2
full name (n), <email_n@domain>, Codalab_user_n

1. SUMMARY OF CONTRIBUTIONS

- *Include a short description of the model architecture[1] (you can use images, text or tables) and a brief justification about your decision (why have you selected/defined it?). Detail the changes you made in the model provided with the starting-kit, if any.*
- *Detail how you have addressed the bias mitigation problem given the different attributes (**age**, **gender**, **ethnicity** and **expression**) and task requirements. For instance, have you prioritized any attribute or considered them jointly?*

1.1 Data augmentation strategy
*Describe in detail the proposed solution. How was the **data augmentation** strategy applied? Justify your decisions.*

1.2 Custom loss strategy
*Describe in detail the proposed solution. How was the **custom loss** strategy applied? Justify your decisions.*

1.2 Training strategy
Detail the training strategy with a brief justification about your decisions. How were the models trained? Have you used any pre-trained model? What optimizer was used? How have you defined the best set of hyperparameters? Etc.

2. EXPERIMENTS AND RESULTS
Describe in detail the different experiments you have defined and present the different results you have obtained, with analysis and discussion of the results. It should include:

- *A clear description of the experiments and their goals. That is, what did you want to analyze and with what objectives? For example, compare different model architectures, hyperparameters, training strategies, pre-trained models, with/without data augmentation, etc? You can define and run as many experiments you need and select the ones you consider more relevant to be included in the report, taking into account you will only have **4 pages in total**. You can use Tables to compare the different results and experiments (progressively), images or graphs.*
- ***One experiment comparing the results obtained when using data augmentation only vs. custom loss (without data augmentation) must be included and discussed.***
- *Explainable models can be a plus (e.g., saliency maps)*

[1] We recommend you to use the same model architecture on all tasks, so that the results can be easily compared (but you are free to use different models in case you want).

*Next, we illustrate how to include **Tables** and **Figures** in your report as well as how you could start a discussion around the results. For instance, you can analyze and compare model X with model Y and comment why model Y obtained overall better results. **Don't forget that the main goal is to maximize accuracy while minimizing (all) the bias scores (i.e., avoid just focusing on minimizing MAE).***

"Table 1 shows obtained results of experiment A. As it can be seen, model X obtained higher Mean Absolute Error. However, it was able to lower down the bias score associated with the Expression attribute compared to model Y, which may be justified by the fact that our strategy prioritized this attribute based on (...)"

Table 1: Experiment A. Hypothetically comparing the results obtained for model X and Y while keeping other variables fixed (i.e., learning rate and training strategy).

Model	Learning rate	Training strategy	Gender bias	Expression bias	Ethnicity bias	Age bias	MAE
X	1e-5	2	0.628002	0.819692	2.447512	8.988890	11.141265
Y	1e-5	2	0.293883	1.027783	2.130759	7.460989	10.975767

"In Figure 1, we illustrate the training curves of strategy 1. As it can be observed, (...)"

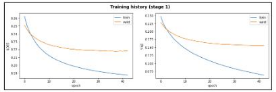


Figure 1: illustrative example of training curve.

In the end, based on your experiments, results and analysis, you should be able to suggest, whenever possible, what strategy better achieved our goal.

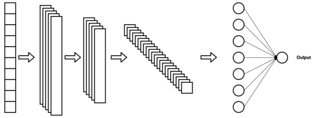
3. FINAL REMARKS
Draw your final remarks, conclusions and findings. For instance, you can comment why you believed strategy Y worked better than X for the problem at hand (and goal), what you believe could make a difference as suggestions for future work, etc.

----- The Report Document MUST not exceed 3 Pages -----
(keep the same font size and margins of the template)

- *First, download this document so that you can edit it.*
- *When you are done, save your report document as a ".pdf" file.*
- *Zip your report document (".pdf file") with the associated "colab.ipynb" file and deliver it as one single zip file through eCampus.*

PAGE LIMIT = 3

What should be in the Report?



1. SUMMARY OF CONTRIBUTIONS

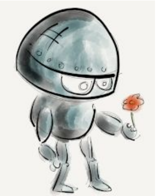
- *Include a short description of the model architecture[1] (you can use images, text or tables) and a brief justification about your decision (why have you selected/defined it?). Detail the changes you made in the model provided with the starting-kit, if any.*
- *Detail how you have addressed the bias mitigation problem given the different attributes (**age**, **gender**, **ethnicity** and **expression**) and task requirements. For instance, have you prioritized any attribute or considered them jointly?*

1.1. Data augmentation strategy (in the case of task 1 or 3)

*Describe in detail the proposed solution. How was the **data augmentation** strategy applied? Justify your decisions.*

1.2. Custom loss strategy (in the case of task 2 or 3)

*Describe in detail the proposed solution. How was the **custom loss** strategy applied? Justify your decisions.*



1.3. Training strategy

Detail the training strategy with a brief justification about your decisions. How were the models trained? Have you used any pre-trained model? What optimizer was used? How have you defined the best set of hyperparameters? Etc.

[1] We recommend you to use the same model architecture on all tasks, so that the results can be easily compared (but you are free to use different models in case you want).

What should be in the Report?

2. EXPERIMENTS AND RESULTS

Describe in detail the different experiments you have defined and present the different results you have obtained, with analysis and discussion of the results. It should include:

- *A clear description of the experiments and their goals. That is, what did you want to analyze and with what objectives? For example, compare different model architectures, hyperparameters, training strategies, pre-trained models, with/without data augmentation, etc? You can define and run as many experiments you need and select the ones you consider more relevant to be included in the report, taking into account you will only have* ***4 pages in total****. You can use Tables to compare the different results and experiments (progressively), images or graphs.*
- ***One experiment comparing the results obtained when using data augmentation only vs. custom loss (without data augmentation) must be included and discussed.***
- *Explainable models can be a plus (e.g., saliency maps)*

What should be in the Report?

2. EXPERIMENTS AND RESULTS

Describe in detail the different experiments you have defined and present the different results you have obtained, with analysis and discussion of the results. It should include:

- *A clear description of the experiments and their goals. That is, what did you want to analyze and with what objectives? For example, compare different model architectures, hyperparameters, training strategies, pre-trained models, with/without data augmentation, etc? You can define and run as many experiments you need and select the ones you consider more relevant to be included in the report, taking into account you will only have* ***3 pages in total****. You can use Tables to compare the different results and experiments (progressively), images or graphs.*
- ***Mandatory for task 1:*** *compare the results you obtained using the baseline model (starting-kit - after 2nd stage of training without data augmentation) with the best results you obtained using the proposed model with data augmentation.*
- ***Mandatory for task 2:*** *compare the results you obtained using the baseline model (starting-kit - after 2nd stage of training without data augmentation) with the best results you obtained using the proposed model with custom loss (in this case, without using data augmentation). Optional (can be a plus): also include the analysis and discussion the results you obtained on task 1.*
- ***Mandatory for (the optional) task 3:*** *compare the results you obtained using the baseline model (starting-kit - after 2nd stage of training without data augmentation) with the best results you obtained on task 1, task 2 and the optional task 3.*

What should be in the Report?

3. FINAL REMARKS

Draw your final remarks, conclusions and findings. For instance, you can comment why you believed strategy Y worked better than X for the problem at hand (and goal), what you believe could make a difference as suggestions for future work, etc.

----- The Report Document MUST not exceed 3 Pages -----

(keep the same font size and margins of the template)

What should be in the Report?

3. FINAL REMARKS

Draw your final remarks, conclusions and findings. For instance, you can comment why you believed strategy Y worked better than X for the problem at hand (and goal), what you believe could make a difference as suggestions for future work, etc.

----- The Report Document MUST not exceed 3 Pages -----

(keep the same font size and margins of the template)

Don't forget that the main goal is to **maximize accuracy** while **minimizing (ALL) the bias scores**.

What should be in the Report?

3. FINAL REMARKS

Draw your final remarks, conclusions and findings. For instance, you can comment why you believed strategy Y worked better than X for the problem at hand (and goal), what you believe could make a difference as suggestions for future work, etc.

----- The Report Document MUST not exceed 3 Pages -----

(keep the same font size and margins of the template)

Don't forget that the main goal is to **maximize accuracy** while **minimizing (ALL) the bias scores**.

That is, **don't centralize your discussions around accuracy only!**

… avoid

- Applying minor changes to the starting-kit
    - **Model**
        - Using the exact same model with different hyperparameters → "be creative"
    - **Augmentation**
        - Applying the same transformations or minor changes → "be creative"
        - Considering the same attribute (age only) → "be creative"
    - **Custom Loss**
        - Applying the same weights / strategy → "be creative"
        - Considering the same attribute (age only) → "be creative"
    - **Training Strategy**
        - Using the same training strategy → "be creative"

Important

We expect to receive a **clear and good discussion around clearly defined experiments**, with **Tables** and **visualizations** (training curves, augmented data examples, etc) **+ "surprise us"**

The Colab code will complement the report document. That is, we will pay more attention to the report document, but we also expect a clear and well documented code, where we can check the final implementation, experiments and obtained results.

Remember: The **Rank position** on Codalab **WON'T be considered** for the evaluation!

Evaluation

- **Report + Colab File: List of items** and ***achievement levels* + Creativity**
    - Task goal (e.g., data augmentation or custom loss) will have high weight

Level of achievement

✅	Played with **hyperparameters**?	❎ Low	❎ Mid	✅ High
✅	Played with different **backbones** (optional)?	❎ Low	❎ Mid	✅ High
✅	Played with the **layers** of the Net?	❎ Low	❎ Mid	✅ High
✅	Performed data **augmentation**?	❎ Low	❎ Mid	✅ High
✅	**Presentation** of the results	❎ Low	❎ Mid	✅ High
✅	**Analysis/discussion** of the results	❎ Low	❎ Mid	✅ High
✅	...	❎ ...	❎ ...	✅ ...

Schedule

- Problem definition class: **28/02**/2023
- Task 1 deadline: **07/03**/2023
- Task 2 deadline: **14/03**/2023
- Control session: **17/03**/2023
    - Detailed feedback and discussion around the delivered tasks, that may be useful to improve the optional task 3.
- Optional Task 3 deadline: **21/03**/2023

Colab demo