Prediction of the clients age for the grocery store

Content

Project Description

Grocery store wants to use computer vision system for the clients photos processing for defenition of their ages. It would help the store to analyze the purchases and suggest the relevant products for the exact clients age group, in addition it would check the cashier behavior during the sell of the age restricted products in accordance with compliance with law. For the realization of such tool it's required to train a model which predicts the approximate age of the client by the photo.

Main task - using copmuter vision tools an machine learning to predict the buyer's age in the shop.

The following taks to be performed:

- data import and explorational analysis;
- prepare the code and upload it on GPU
- upload the code on GPU for prediction calculation;
- to analyze the the results of prediction;
- draw a conclsuion

Data import and expolatory analysis

Import of libraries and data load

```
In [1]: from tensorflow.keras.datasets import fashion_mnist
        from tensorflow.keras.layers import Dense
        from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Conv2D, Flatten, Dense, MaxPooling2D
        from tensorflow.keras.optimizers import Adam
        from tensorflow.keras.preprocessing.image import ImageDataGenerator
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [2]: datagen = ImageDataGenerator(validation_split = 0.25,
                                     rescale = 1/255)
```

```
In [3]: train_datagen_flow = datagen.flow_from_directory('/datasets/faces/',
                                                         target_size = (150,150),
                                                         batch_size = 16,
                                                         class_mode = 'sparse',
                                                         seed = 12345)

        Found 7591 images belonging to 1 classes.
```

```
In [4]: features, target = next(train_datagen_flow)
```

```
In [5]: target

Out[5]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
              dtype=float32)
```

```
In [6]: df_age = pd.read_csv('/datasets/faces/labels.csv')
```

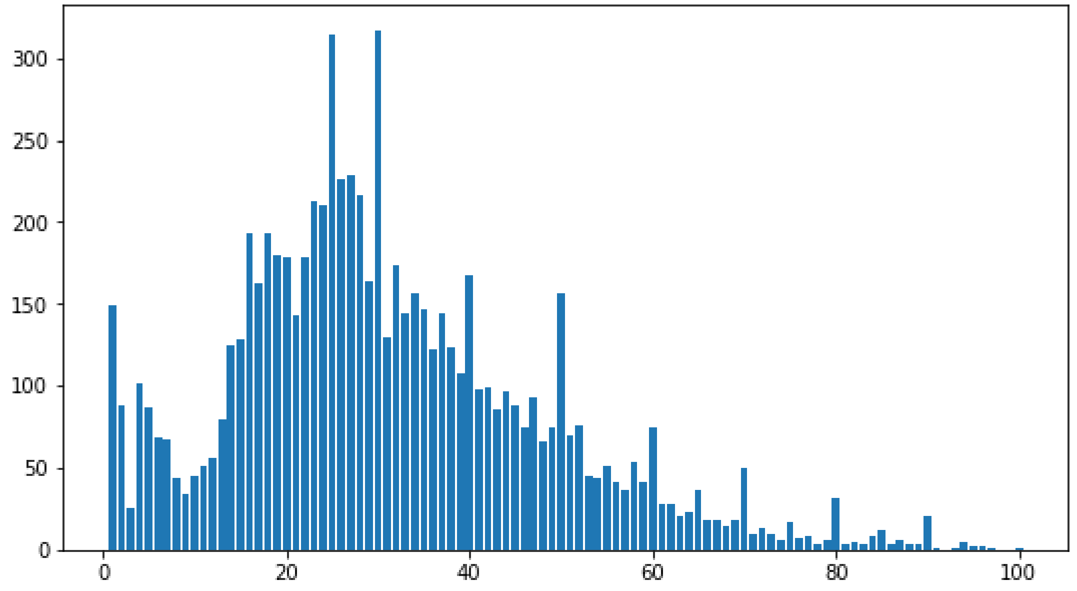
Display the information on the quantity of records in dataset, analyse the statistic of data and display a few pictures of buyers

```
In [7]: df_age.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7591 entries, 0 to 7590
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   file_name  7591 non-null   object
 1   real_age   7591 non-null   int64
dtypes: int64(1), object(1)
memory usage: 118.7+ KB
```

In [8]:
```
df_count_by_age = df_age.groupby('real_age').count().sort_values(by = 'file_name', ascending = False).reset_index()
df_count_by_age = df_count_by_age.rename(columns={"file_name": "count"})
plt.figure(figsize=(9, 5))
plt.bar(df_count_by_age['real_age'],df_count_by_age['count'])
```

Out[8]:
```
<BarContainer object of 97 artists>
```



In [9]:
```
df_count_by_age[df_count_by_age['real_age'] == df_count_by_age['real_age'].min()]
```

Out[9]:

	real_age	count
18	1	149

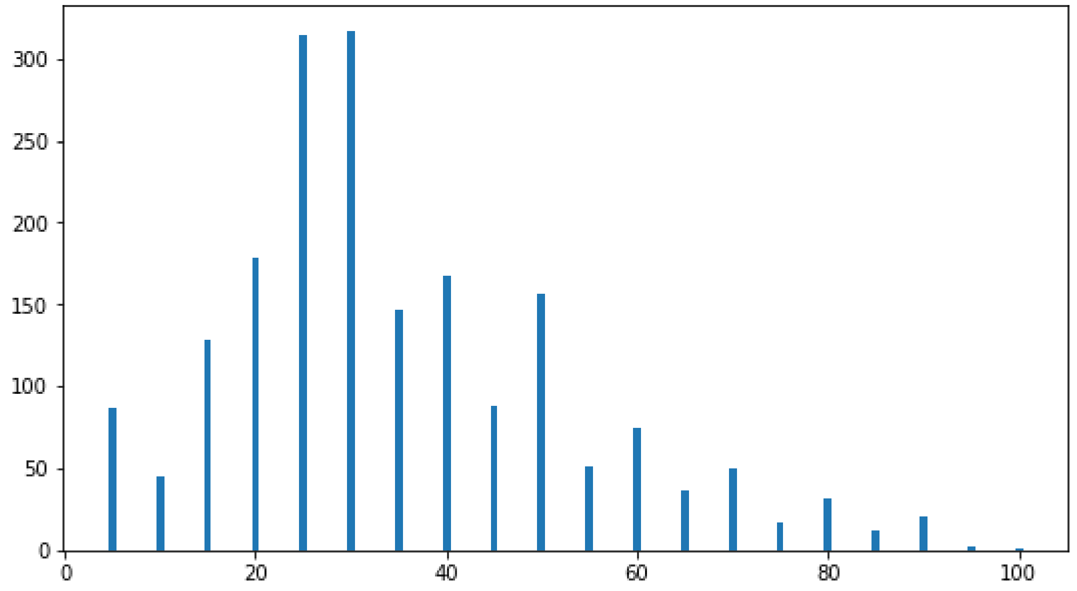
```
In [10]: df_count_by_age[df_count_by_age['real_age'] == df_count_by_age['real_age'].max()]
```

Out[10]:

	real_age	count
96	100	1

```
In [11]: df_count_by_age_five = df_count_by_age[df_count_by_age['real_age'] % 5 == 0].reset_index(drop =True)
         plt.figure(figsize=(9, 5))
         plt.bar(df_count_by_age_five['real_age'],df_count_by_age_five['count'])
```

```
Out[11]: <BarContainer object of 20 artists>
```



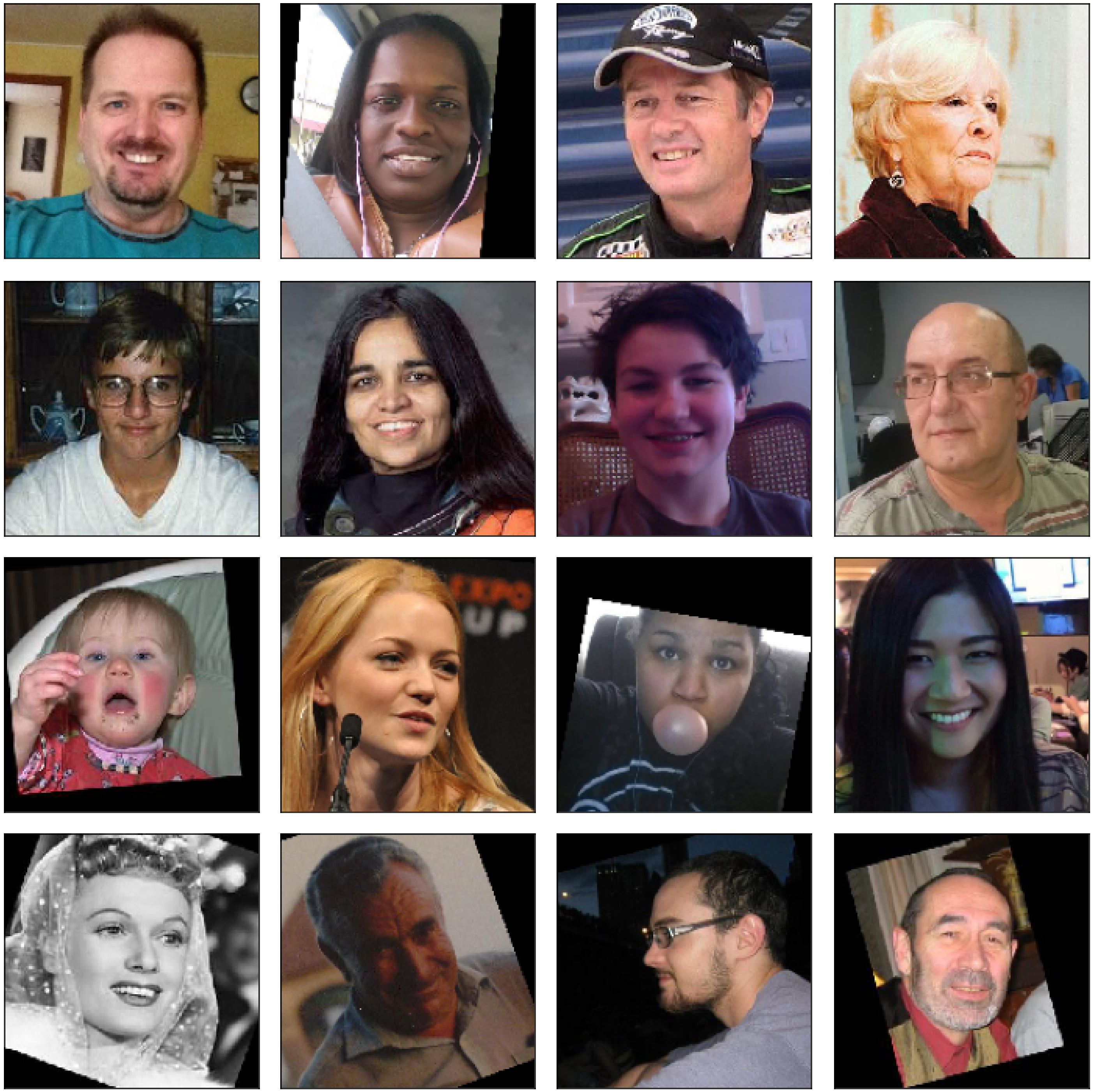
```
In [12]: df_count_by_age_five.sort_values(by = 'real_age').reset_index(drop =True)
```

Out[12]:

	real_age	count
0	5	87
1	10	45
2	15	128
3	20	178
4	25	315
5	30	317
6	35	147
7	40	167
8	45	88
9	50	156
10	55	51
11	60	75
12	65	36
13	70	50
14	75	17
15	80	31
16	85	12
17	90	21
18	95	2
19	100	1

In [13]:

```
fig = plt.figure(figsize=(10,10))
for i in range(16):
    fig.add_subplot(4, 4, i+1)
    plt.imshow(features[i])
    plt.xticks([])
    plt.yticks([])
    plt.tight_layout()
```



Exploratoty analysis conclusion:

- Total quanity of records in data is 7591;
- The age of most part of buyers is from 15 to 40 years;
- minimal age - 1 (149 photos), maximum - 100 (1 photo);
- Имеются выбросы по количеству фото в районе 1 года, 25, 35, 40 50 и 80 лет (скорее всего фото по случаю юбилея);
- Buyer's picture successfuly displayed and size unified;
- Tha sample has colorfull and black and white photos;
- On the photo the people of different age, sex and nationality are represented;
- Photos were made from the different angles and positiona of camera;
- Photos has a forign object in the frame (such as glasses, chewing gum, microphone, hand, veil, etc;)
- Photos were made in different surrounding area and different type of lighting;
- Some of the photos are rotated on different angle and don't fit 150*150 frame.

Code for model trainig and result preduction using GPU

In [14]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Flatten, Dense, MaxPooling2D, GlobalAveragePooling2D,Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.resnet50 import ResNet50
import numpy as np
import pandas as pd
```

In [15]:
```
def load_train(path):
    train_datagen = ImageDataGenerator(
        validation_split=0.25,
        rescale=1./255,
        horizontal_flip=True
    )
    data_frame = pd.read_csv(path + 'labels.csv')
    train_datagen_flow = train_datagen.flow_from_dataframe(dataframe = data_frame,
                                                           directory = (path + 'final_files/'),
                                                           x_col = 'file_name' ,
                                                         y_col = 'real_age',
                                                         target_size = (224,224),
                                                         batch_size = 64,
                                                         class_mode = 'other',
                                                         subset='training',
                                                         seed = 12345)
    return train_datagen_flow
```

In [16]:
```
train_datagen_flow = load_train('/datasets/faces/')
```

```
--- Logging error ---
Traceback (most recent call last):
  File "/opt/conda/lib/python3.9/logging/__init__.py", line 1083, in emit
    msg = self.format(record)
  File "/opt/conda/lib/python3.9/logging/__init__.py", line 927, in format
    return fmt.format(record)
  File "/opt/conda/lib/python3.9/logging/__init__.py", line 663, in format
    record.message = record.getMessage()
  File "/opt/conda/lib/python3.9/logging/__init__.py", line 367, in getMessage
    msg = msg % self.args
TypeError: not all arguments converted during string formatting
Call stack:
  File "/opt/conda/lib/python3.9/runpy.py", line 197, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "/opt/conda/lib/python3.9/runpy.py", line 87, in _run_code
    exec(code, run_globals)
  File "/opt/conda/lib/python3.9/site-packages/ipykernel_launcher.py", line 16, in <module>
    app.launch_new_instance()
  File "/opt/conda/lib/python3.9/site-packages/traitlets/config/application.py", line 845, in launch_instance
    app.start()
  File "/opt/conda/lib/python3.9/site-packages/ipykernel/kernelapp.py", line 668, in start
    self.io_loop.start()
  File "/opt/conda/lib/python3.9/site-packages/tornado/platform/asyncio.py", line 199, in start
    self.asyncio_loop.run_forever()
  File "/opt/conda/lib/python3.9/asyncio/base_events.py", line 596, in run_forever
    self._run_once()
  File "/opt/conda/lib/python3.9/asyncio/base_events.py", line 1890, in _run_once
    handle._run()
  File "/opt/conda/lib/python3.9/asyncio/events.py", line 80, in _run
    self._context.run(self._callback, *self._args)
  File "/opt/conda/lib/python3.9/site-packages/ipykernel/kernelbase.py", line 456, in dispatch_queue
    await self.process_one()
  File "/opt/conda/lib/python3.9/site-packages/ipykernel/kernelbase.py", line 445, in process_one
    await dispatch(*args)
  File "/opt/conda/lib/python3.9/site-packages/ipykernel/kernelbase.py", line 352, in dispatch_shell
    await result
  File "/opt/conda/lib/python3.9/site-packages/ipykernel/kernelbase.py", line 647, in execute_request
    reply_content = await reply_content
  File "/opt/conda/lib/python3.9/site-packages/ipykernel/ipkernel.py", line 335, in do_execute
    res = shell.run_cell(code, store_history=store_history, silent=silent)
  File "/opt/conda/lib/python3.9/site-packages/ipykernel/zmqshell.py", line 532, in run_cell
    return super(ZMQInteractiveShell, self).run_cell(*args, **kwargs)
  File "/opt/conda/lib/python3.9/site-packages/IPython/core/interactiveshell.py", line 2898, in run_cell
```

```
    result = self._run_cell(
  File "/opt/conda/lib/python3.9/site-packages/IPython/core/interactiveshell.py", line 2944, in _run_cell
    return runner(coro)
  File "/opt/conda/lib/python3.9/site-packages/IPython/core/async_helpers.py", line 68, in _pseudo_sync_runner
    coro.send(None)
  File "/opt/conda/lib/python3.9/site-packages/IPython/core/interactiveshell.py", line 3169, in run_cell_async
    has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
  File "/opt/conda/lib/python3.9/site-packages/IPython/core/interactiveshell.py", line 3361, in run_ast_nodes
    if (await self.run_code(code, result,  async_=asy)):
  File "/opt/conda/lib/python3.9/site-packages/IPython/core/interactiveshell.py", line 3441, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "/tmp/ipykernel_534/3953403445.py", line 1, in <module>
    train_datagen_flow = load_train('/datasets/faces/')
  File "/tmp/ipykernel_534/755795560.py", line 8, in load_train
    train_datagen_flow = train_datagen.flow_from_dataframe(dataframe = data_frame,
  File "/opt/conda/lib/python3.9/site-packages/keras/preprocessing/image.py", line 1107, in flow_from_dataframe
    tf_logging.warning(
  File "/opt/conda/lib/python3.9/site-packages/tensorflow/python/platform/tf_logging.py", line 178, in warning
    get_logger().warning(msg, *args, **kwargs)
Message: '`class_mode` "other" is deprecated, please use `class_mode` "raw".'
Arguments: (<class 'DeprecationWarning'>,)
Found 5694 validated image filenames.
```

In [17]:
```python
def load_test(path):
    test_datagen = ImageDataGenerator(
        validation_split=0.25,
        rescale=1./255)

    data_frame = pd.read_csv(path + 'labels.csv')
    test_datagen_flow = test_datagen.flow_from_dataframe(dataframe=data_frame,
                                                         directory = path + 'final_files/',
                                                         x_col = 'file_name' ,
                                                         y_col = 'real_age',
                                                         target_size = (224,224),
                                                         batch_size = 64,
                                                         class_mode = 'other',
                                                         subset='validation',
                                                         seed = 12345)

    return test_datagen_flow
```

In [18]:
```python
def create_model(input_shape):
    backbone = ResNet50(input_shape=input_shape,
                    weights='/datasets/keras_models/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5',
                    include_top=False)

    model = Sequential()
    model.add(backbone)
    model.add(GlobalAveragePooling2D())
    model.add(Dense(1, activation = 'relu'))
    optimizer = Adam(lr = 0.00005)
    model.compile(optimizer= optimizer, loss='mean_squared_error',
                  metrics=['mean_absolute_error'])
    return model
```

In [19]:
```
def train_model(model, train_data, test_data, batch_size=None, epochs=30,
                steps_per_epoch=None, validation_steps=None):

    model.fit(train_data,
              validation_data=(test_data),
              batch_size=batch_size,
              epochs=epochs,
              steps_per_epoch=steps_per_epoch,
              validation_steps=validation_steps,
              verbose=2, shuffle=True)

    return model
```

GPU prediction results

2022-12-12 17:52:09.503553: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libnvinfer.so.6 2022-12-12 17:52:09.505339: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libnvinfer_plugin.so.6 2022-12-12 17:52:10.406111: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcuda.so.1 2022-12-12 17:52:10.416183: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1555] Found device 0 with properties: pciBusID: 0000:8b:00.0 name: Tesla V100-SXM2-32GB computeCapability: 7.0 coreClock: 1.53GHz coreCount: 80 deviceMemorySize: 31.75GiB deviceMemoryBandwidth: 836.37GiB/s 2022-12-12 17:52:10.416278: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudart.so.10.1 2022-12-12 17:52:10.416313: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcublas.so.10 2022-12-12 17:52:10.418282: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcufft.so.10 2022-12-12 17:52:10.418740: I

```
tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcurand.so.10 2022-12-12 17:52:10.420899:
I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcusolver.so.10 2022-12-12
17:52:10.422093: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcusparse.so.10 2022-12-
12 17:52:10.422181: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudnn.so.7 2022-12-
12 17:52:10.426632: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1697] Adding visible gpu devices: 0 Using TensorFlow backend.
Found 5694 validated image filenames. Found 1897 validated image filenames. 2022-12-12 17:52:10.605005: I
tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
AVX512F FMA 2022-12-12 17:52:10.611753: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 2100000000 Hz 2022-12-12
17:52:10.612279: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x4a7dcb0 initialized for platform Host (this does not guarantee
that XLA will be used). Devices: 2022-12-12 17:52:10.612295: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host,
Default Version 2022-12-12 17:52:10.771186: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x40ddfd0 initialized for platform
CUDA (this does not guarantee that XLA will be used). Devices: 2022-12-12 17:52:10.771226: I tensorflow/compiler/xla/service/service.cc:176]
StreamExecutor device (0): Tesla V100-SXM2-32GB, Compute Capability 7.0 2022-12-12 17:52:10.773613: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1555] Found device 0 with properties: pciBusID: 0000:8b:00.0 name: Tesla V100-SXM2-
32GB computeCapability: 7.0 coreClock: 1.53GHz coreCount: 80 deviceMemorySize: 31.75GiB deviceMemoryBandwidth: 836.37GiB/s 2022-12-12
17:52:10.773694: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudart.so.10.1 2022-12-
12 17:52:10.773705: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcublas.so.10 2022-
12-12 17:52:10.773736: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcufft.so.10 2022-
12-12 17:52:10.773747: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcurand.so.10
2022-12-12 17:52:10.773756: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
libcusolver.so.10 2022-12-12 17:52:10.773765: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic
library libcusparse.so.10 2022-12-12 17:52:10.773773: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened
dynamic library libcudnn.so.7 2022-12-12 17:52:10.778389: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1697] Adding visible gpu
devices: 0 2022-12-12 17:52:10.778462: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
libcudart.so.10.1 2022-12-12 17:52:11.171660: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1096] Device interconnect StreamExecutor
with strength 1 edge matrix: 2022-12-12 17:52:11.171718: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1102] 0 2022-12-12
17:52:11.171726: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] 0: N 2022-12-12 17:52:11.176281: W
tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc:39] Overriding allow_growth setting because the TF_FORCE_GPU_ALLOW_GROWTH
environment variable is set. Original config value was 0. 2022-12-12 17:52:11.176339: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1241] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 10240
MB memory) -> physical GPU (device: 0, name: Tesla V100-SXM2-32GB, pci bus id: 0000:8b:00.0, compute capability: 7.0)
```

```
<class 'tensorflow.python.keras.engine.sequential.Sequential'> WARNING:tensorflow:sample_weight modes were coerced from ... to
['...'] WARNING:tensorflow:sample_weight modes were coerced from ... to
['...'] Train for 89 steps, validate for 30 steps Epoch 1/30 2022-12-12 17:52:22.519506: I
tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcublas.so.10 2022-12-12 17:52:22.848163:
I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudnn.so.7 2022-12-12 17:52:25.056486:
W tensorflow/core/common_runtime/bfc_allocator.cc:243] Allocator (GPU_0_bfc) ran out of memory trying to allocate 2.55GiB with
freed_by_count=0. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memory were
available. 2022-12-12 17:52:25.056556: W tensorflow/core/common_runtime/bfc_allocator.cc:243] Allocator (GPU_0_bfc) ran out of memory trying
to allocate 2.55GiB with freed_by_count=0. The caller indicates that this is not a failure, but may mean that there could be performance gains if
more memory were available. 89/89 - 49s - loss: 431.8952 - mean_absolute_error: 15.4452 - val_loss: 1089.6798 - val_mean_absolute_error:
28.4695 Epoch 2/30 89/89 - 41s - loss: 84.5932 - mean_absolute_error: 6.8504 - val_loss: 1099.2107 - val_mean_absolute_error: 28.6304 Epoch
3/30 89/89 - 41s - loss: 51.6925 - mean_absolute_error: 5.4030 - val_loss: 1121.2040 - val_mean_absolute_error: 29.0198 Epoch 4/30 89/89 - 41s -
loss: 35.3124 - mean_absolute_error: 4.4696 - val_loss: 1100.4054 - val_mean_absolute_error: 28.6951 Epoch 5/30 89/89 - 41s - loss: 26.0578 -
mean_absolute_error: 3.8490 - val_loss: 841.8695 - val_mean_absolute_error: 24.1768 Epoch 6/30 89/89 - 41s - loss: 19.9586 -
mean_absolute_error: 3.3780 - val_loss: 493.3221 - val_mean_absolute_error: 17.1029 Epoch 7/30 89/89 - 41s - loss: 15.7746 -
mean_absolute_error: 2.9907 - val_loss: 199.3575 - val_mean_absolute_error: 10.3596 Epoch 8/30 89/89 - 41s - loss: 13.4835 -
mean_absolute_error: 2.8050 - val_loss: 122.4511 - val_mean_absolute_error: 8.5759 Epoch 9/30 89/89 - 42s - loss: 11.5009 - mean_absolute_error:
2.5658 - val_loss: 88.2129 - val_mean_absolute_error: 7.0944 Epoch 10/30 89/89 - 41s - loss: 10.0387 - mean_absolute_error: 2.4053 - val_loss:
92.2139 - val_mean_absolute_error: 7.2421 Epoch 11/30 89/89 - 40s - loss: 9.0945 - mean_absolute_error: 2.3081 - val_loss: 68.6850 -
val_mean_absolute_error: 6.2210 Epoch 12/30 89/89 - 41s - loss: 8.6912 - mean_absolute_error: 2.2300 - val_loss: 72.1850 -
val_mean_absolute_error: 6.3667 Epoch 13/30 89/89 - 40s - loss: 8.0173 - mean_absolute_error: 2.1573 - val_loss: 71.4363 -
val_mean_absolute_error: 6.2717 Epoch 14/30 89/89 - 40s - loss: 8.0521 - mean_absolute_error: 2.1251 - val_loss: 67.7193 -
val_mean_absolute_error: 6.2128 Epoch 15/30 89/89 - 41s - loss: 7.4678 - mean_absolute_error: 2.0929 - val_loss: 69.4032 -
val_mean_absolute_error: 6.1821 Epoch 16/30 89/89 - 41s - loss: 6.9028 - mean_absolute_error: 2.0164 - val_loss: 74.9971 -
val_mean_absolute_error: 6.5819 Epoch 17/30 89/89 - 41s - loss: 6.3498 - mean_absolute_error: 1.9277 - val_loss: 66.3816 -
val_mean_absolute_error: 6.0624 Epoch 18/30 89/89 - 40s - loss: 5.7586 - mean_absolute_error: 1.8381 - val_loss: 67.5881 -
val_mean_absolute_error: 6.1760 Epoch 19/30 89/89 - 41s - loss: 5.5005 - mean_absolute_error: 1.7803 - val_loss: 79.5287 -
val_mean_absolute_error: 6.5636 Epoch 20/30 89/89 - 41s - loss: 5.6725 - mean_absolute_error: 1.8155 - val_loss: 65.8956 -
val_mean_absolute_error: 6.1417 Epoch 21/30 89/89 - 41s - loss: 4.6942 - mean_absolute_error: 1.6333 - val_loss: 67.3597 -
val_mean_absolute_error: 6.1864 Epoch 22/30 89/89 - 41s - loss: 4.4607 - mean_absolute_error: 1.5931 - val_loss: 66.3814 -
val_mean_absolute_error: 6.1737 Epoch 23/30 89/89 - 41s - loss: 4.4770 - mean_absolute_error: 1.5881 - val_loss: 69.3931 -
```

```
val_mean_absolute_error: 6.2105 Epoch 24/30 89/89 - 41s - loss: 4.0751 - mean_absolute_error: 1.5254 - val_loss: 63.7006 -
val_mean_absolute_error: 5.9689 Epoch 25/30 89/89 - 41s - loss: 3.7874 - mean_absolute_error: 1.4649 - val_loss: 64.4568 -
val_mean_absolute_error: 6.0407 Epoch 26/30 89/89 - 41s - loss: 3.4774 - mean_absolute_error: 1.3988 - val_loss: 64.6785 -
val_mean_absolute_error: 6.0982 Epoch 27/30 89/89 - 40s - loss: 3.4915 - mean_absolute_error: 1.4067 - val_loss: 63.3570 -
val_mean_absolute_error: 5.9098 Epoch 28/30 89/89 - 40s - loss: 3.4148 - mean_absolute_error: 1.3910 - val_loss: 64.2029 -
val_mean_absolute_error: 5.9535 Epoch 29/30 89/89 - 41s - loss: 3.5640 - mean_absolute_error: 1.3948 - val_loss: 67.3844 -
val_mean_absolute_error: 6.0585 Epoch 30/30 89/89 - 41s - loss: 3.5523 - mean_absolute_error: 1.4133 - val_loss: 63.4697 -
val_mean_absolute_error: 5.8911 WARNING:tensorflow:sample_weight modes were coerced from ... to
['...'] 30/30 - 10s - loss: 63.4697 - mean_absolute_error: 5.8911 Test MAE: 5.8911
```

General Conclusions

1) For execution of the task the exploratory analysis was successfully performed;

2) The code for GPU was prepared and uploded.

3) The code architecture was builded on ResNet50 (backbone, w/o top) with pretrained weight without freezing and with GlobalAverage2Dpooling with and 1 layer of one neuron anf relu activation fuction;

4) Batch size - 64 ;

5) Optimiser - Адам, learning rate - 0,00005;

6) Epochs quantity - 30;

7) Obtained results - 5,89 MAE on test sample, model overtraining was not occured.

```
In [ ]:
```