

## СОЗДАНИЕ ПРОГРАММЫ ДЛЯ ОЦЕНКИ ОБРАЗА

Пристинский Я.А., ученик 10 «А» класса

Научный руководитель: Гараничева С.В., учитель информатики  
Муниципальное бюджетное общеобразовательное учреждение «Лицей №23»  
г. Кемерово

### Введение/проблематика

Не знаю как вы, но я каждое утро метаюсь по комнате и не могу выбрать одежду, в которой собираюсь выйти из дома. Вся проблема заключается в том, что для данного выбора характерно сочетание нескольких факторов, которое довольно трудно соблюдать, когда ты опаздываешь или все еще находишься частично во сне. Именно в такие моменты понимаешь, что просто необходимо сбросить лишние проблемы с себя и переложить их решение на другой, максимально объективный мозг – «искусственный мозг».

Именно после подобно размышления в голове мне пришло в голову сделать программу, позволяющую упростить выбор уже имеющейся у меня одежды.

### Цель работы

Целью работы является создание программы, оценивающей образ пользователя по сделанной им фотографии на основе пола пользователя, времени года и выбранной пользователем ситуации.

### Задачи:

1. Поиск и сбор образов, обладающих нужными нам характеристиками и последующая их сортировка;
2. Выбор инструментов;
3. Написание кода нейронной сети, основанной на компьютерном зрении;
4. Обучение написанной сети;
5. Проверка полученных результатов.

### Задача №1

Поиск образов для датасета был сделан с помощью ресурсов Яндекс.Картинки, Google Images и Pinterest. Все образы обладали собственными характеристиками, такими как пол(мужской, женский), сезон(лето, зима, демисезон) и стиль(спортивный, деловой, вечерний и повседневный). В итоге я собрал 952 картинок образов и отсортировал их. В среднем на каждую ситуацию пришлось от 35 до 50 картинок образов. Структура их сортировки показана на рисунке 1.

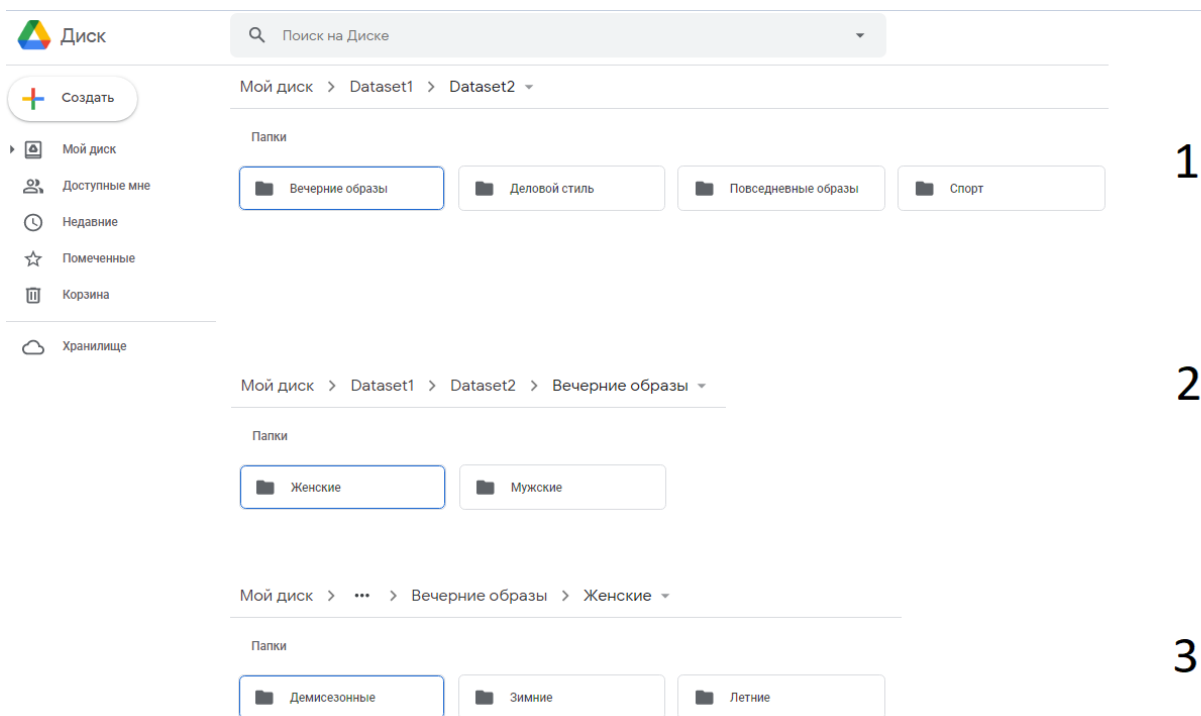


Рисунок 1 - Структура сортировки образов  
**Задача №2**

Как вы уже могли увидеть на рисунке 1, я выбрал среду Google для выполнения работы. Их облачное хранилище позволило хранить датасет и сам проект. Рабочее пространство Google Colab позволило писать код на языке программирования Python и взаимодействовать с другими сервисами Google. Также в процессе работы пришлось использовать Google Tabs для работы с таблицами.

### **Задачи №3 и 4**

Скриншоты кода смотрите в Приложении 1.

### **Задача №5**

На следующих рисунках вы можете видеть как я вывожу произвольные картинки образов из тестового множества и смотрю как справилась программа со своим предсказанием. Понять как она справилась довольно просто. В квадратных скобках после значения True показывается правильность образа для выбранных условий. То есть если в квадратных скобках после True цифра 1, то образ правильный, если 0 – неправильный. А само предсказание правильности образа для выбранных характеристик показано после значения Pred. Числовое значение может варьироваться от 0 до 1, где 1 – это 100%-ная уверенность сети в правильности подбора образа к представленным ей характеристикам.

Проверка полученной нейронной сети

```
[ ] 1 index = 30
2 print(
3     'True', array_ok_test[index],
4     'Pred %.3f' % classifier.predict([
5         np.expand_dims(images_array_test[index] / 255., axis=0),
6         np.expand_dims(additional_X_test[0], axis=0)
7     ])[0]
8 )
9
10 print(dataset.columns)
11 print(additional_X_test[index])
12 plt.imshow(images_array_test[index]);
```

True [1] Pred 0.797  
 Index(['Unnamed: 0', 'Женские', 'Мужские', 'Демисезонные', 'Зимние', 'Летние',  
 'Вечерние образы', 'Деловой стиль', 'Повседневные образы', 'Спорт',  
 'filename', 'filepath', 'ok'],  
 dtype='object')

[1 0 0 1 0 0 0 1 0]



```
1 index = 65
2 print(
3     'True', array_ok_test[index],
4     'Pred %.3f' % classifier.predict([
5         np.expand_dims(images_array_test[index] / 255., axis=0),
6         np.expand_dims(additional_X_test[0], axis=0)
7     ])[0]
8 )
9
10 print(dataset.columns)
11 print(additional_X_test[index])
12 plt.imshow(images_array_test[index]);
```

True [1] Pred 0.940  
 Index(['Unnamed: 0', 'Женские', 'Мужские', 'Демисезонные', 'Зимние', 'Летние',  
 'Вечерние образы', 'Деловой стиль', 'Повседневные образы', 'Спорт',  
 'filename', 'filepath', 'ok'],  
 dtype='object')

[1 0 1 0 0 0 0 1 0]




Рисунок 2 – Проверка 1

```
[ ] 1 index = 100
2 print(
3     'True', array_ok_test[index],
4     'Pred %.3f' % classifier.predict([
5         np.expand_dims(images_array_test[index] / 255., axis=0),
6         np.expand_dims(additional_X_test[0], axis=0)
7     ])[0]
8 )
9
10 print(dataset.columns)
11 print(additional_X_test[index])
12 plt.imshow(images_array_test[index]);
```

True [0] Pred 0.310  
 Index(['Unnamed: 0', 'Женские', 'Мужские', 'Демисезонные', 'Зимние', 'Летние',  
 'Вечерние образы', 'Деловой стиль', 'Повседневные образы', 'Спорт',  
 'filename', 'filepath', 'ok'],  
 dtype='object')

[0 1 1 0 0 0 0 0 1]



```
[ ] 1 index = 12
2 print(
3     'True', array_ok_test[index],
4     'Pred %.3f' % classifier.predict([
5         np.expand_dims(images_array_test[index] / 255., axis=0),
6         np.expand_dims(additional_X_test[0], axis=0)
7     ])[0]
8 )
9
10 print(dataset.columns)
11 print(additional_X_test[index])
12 plt.imshow(images_array_test[index]);
```

True [0] Pred 0.179  
 Index(['Unnamed: 0', 'Женские', 'Мужские', 'Демисезонные', 'Зимние', 'Летние',  
 'Вечерние образы', 'Деловой стиль', 'Повседневные образы', 'Спорт',  
 'filename', 'filepath', 'ok'],  
 dtype='object')

[1 0 0 1 0 0 0 1 0]




Рисунок 3 – Проверка 2

```

1 index = 167
2 print(
3     'True', array_ok_test[index],
4     'Pred %.3f' % classifier.predict([
5         np.expand_dims(images_array_test[index] / 255., axis=0),
6         np.expand_dims(additional_X_test[0], axis=0)
7     ])[0]
8 )
9
10 print(dataset.columns)
11 print(additional_X_test[index])
12 plt.imshow(images_array_test[index]);

true [1] Pred 0.845
index[('Unnamed: 0', 'Женские', 'Мужские', 'Демисезонные', 'Зимние', 'Летние',
      'Вечерние образы', 'Деловой стиль', 'Повседневные образы', 'Спорт',
      'filename', 'filepath', 'ok'),
      dtype='object')
0 1 0 1 0 0 1 0 0]
0
20
40
60
80
0 20 40 60 80

```



```

1 test_image = cv2.imread(
2     '/content/drive/MyDrive/Dataset1/Dataset2/look.jpg'
3 )
4 test_image = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
5 test_image = cv2.resize(test_image, (img_size, img_size))
6
7 additional_X_check = list({
8     'Женские': 1, 'Мужские': 0,
9     'Демисезонные': 1, 'Зимние': 0, 'Летние': 0,
10    'Вечерние образы': 0, 'Деловой стиль': 0, 'Повседневные образы': 1, 'Спорт': 0
11 }.values())
12
13 print(
14     'Pred %.3f' % classifier.predict([
15         np.expand_dims(test_image / 255., axis=0),
16         np.expand_dims(additional_X_check, axis=0)
17     ])[0]
18 )
19
20 plt.imshow(test_image);

Pred 0.753
0
20
40
60
80
0 20 40 60 80

```




Рисунок 4 – Проверка 3

Как вы можете видеть нейронная сеть прекрасно справилась с пятью примерами из тестового множества и с одним случайным примером из Интернета.

### Заключение

В итоге работы я создал искусственную нейронную сеть на языке программирования Python, которая полностью справляется со своей работой. В будущем я планирую развивать данный проект путем увеличения числа картинок для обучения и создания мобильного приложения для упрощения работы с нейросетью.

### Приложение 1

```

1 %matplotlib inline
2
3 # импорт библиотек
4 import numpy as np
5 import pandas as pd
6 import matplotlib.pyplot as plt
7 import seaborn as sns
8
9
10
11
12 # загрузка данных с Google Drive
13 from google.colab import drive
14 drive.mount('/content/drive', force_remount=True)
15
16 mounted at /content/drive
17
18
19
20 # загрузка изображений в numpy-массивы
21
22 path = '/content/drive/MyDrive/Dataset1/Dataset2'
23
24 import os
25
26 image_files = []
27
28 for first_level in os.listdir(path):
29     for second_level in os.listdir(path + '/' + first_level):
30         for third_level in os.listdir(path + '/' + first_level + '/' + second_level):
31             for filename in os.listdir(path + '/' + first_level + '/' + second_level + '/' + third_level):
32                 image_files.append({
33                     'style': first_level,
34                     'gender': second_level,
35                     'season': third_level,
36                     'filename': filename,
37                     'filepath': path + '/' + first_level + '/' + second_level + '/' + third_level
38                 })
39
40
41
42
43 datatable = pd.DataFrame(image_files)
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

	style	gender	season	filename	filepath
0	Спорт	Женские	Летние	IMG_0063.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...
1	Спорт	Женские	Летние	IMG_0050.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...
2	Спорт	Женские	Летние	IMG_0052.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...
3	Спорт	Женские	Летние	IMG_0051.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...
4	Спорт	Женские	Летние	IMG_0036.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...
...	...	...	...	...	...
947	Вечерние образы	Мужские	Демисезонные	images (6).jpeg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...
948	Вечерние образы	Мужские	Демисезонные	images (7).jpeg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...
949	Вечерние образы	Мужские	Демисезонные	002_2-341x512.jpg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...
950	Вечерние образы	Мужские	Демисезонные	035_341x512.jpg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...
951	Вечерние образы	Мужские	Демисезонные	images (10).jpeg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...

```

1 datatable[['filepath']].to_excel('paths2.xlsx')
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

	style	gender	season	filename	filepath	ok
0	Спорт	Женские	Летние	IMG_0063.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...	1
1	Спорт	Женские	Летние	IMG_0050.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...	1
2	Спорт	Женские	Летние	IMG_0052.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...	1
3	Спорт	Женские	Летние	IMG_0051.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...	1
4	Спорт	Женские	Летние	IMG_0036.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопт...	1
...	...	...	...	...	...	...
947	Вечерние образы	Мужские	Демисезонные	images (6).jpeg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...	1

Рисунок 5 – Загружаем картинки в таблицу и добавляем столбец ok, все значения которого на данный момент – единицы.

```

1 data = pd.get_dummies(datatable['style']).join(datatable).drop('style', axis='columns')
2 data = pd.get_dummies(data.season).join(data).drop('season', axis='columns')
3 data = pd.get_dummies(data.gender).join(data).drop('gender', axis='columns')
4
5 data['ok'] = 1
6
7 data
    
```

	Женские	Мужские	Демисезонные	Зимние	Летние	Вечерние образы	Деловой стиль	Повседневные образы	Спорт	filename	filepath	ok
0	1	0	0	0	1	0	0	0	1	IMG_0063.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопр...	1
1	1	0	0	0	1	0	0	0	1	IMG_0050.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопр...	1
2	1	0	0	0	1	0	0	0	1	IMG_0052.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопр...	1
3	1	0	0	0	1	0	0	0	1	IMG_0051.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопр...	1
4	1	0	0	0	1	0	0	0	1	IMG_0036.JPG	/content/drive/MyDrive/Dataset1/Dataset2/Cнопр...	1
...	...	...	...	...	...	...	...	...	...	...	...	...
947	0	1	1	0	0	1	0	0	0	images (6).jpeg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...	1
948	0	1	1	0	0	1	0	0	0	images (7).jpeg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...	1
949	0	1	1	0	0	1	0	0	0	002_-2-341x512.jpg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...	1
950	0	1	1	0	0	1	0	0	0	035_-341x512.jpg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...	1
951	0	1	1	0	0	1	0	0	0	images (10).jpeg	/content/drive/MyDrive/Dataset1/Dataset2/Вечер...	1

952 rows x 12 columns

```

[ ] 1 # Скачиваем датасет как таблицу
2 data.to_excel('data3_2.xlsx')
3
4 # Копируем эту таблицу и в копии дублируем все строки, заменяем для дубликатов значение в колонке ok на 0, а также меняем в дубликатах по одной характеристике
    
```

Рисунок 6 – Разбиваем столбцы с характеристиками и скачиваем готовую таблицу, немного изменяем ее

```

1 # Меняем наш старый датасет на новый с дубликатами(его название - "data3_3.xlsx")
2 # Разбиваем все данные в таблице на 3 массива:
3 # images_array, additional_X - хранит все фичи, array_ok - состоит из одной колонки ok(то есть определяет правильность образа для фич, заданных в таблице)
4
5 import cv2
6
7 dataset = pd.read_excel('/content/drive/MyDrive/Dataset1/data3_3.xlsx')
8 print(dataset.columns)
9
10 additional_X = []
11 array_ok = []
12 images_array = None
13
14 img_size = 100
15
16 for index in range(len(dataset)):
17     image_file = dataset.iloc[index]['filepath'] + '/' + dataset.iloc[index]['filename']
18     if index < len(dataset) // 2:
19         image = cv2.imread(image_file)
20
21     try:
22         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
23     except Exception as exception:
24         print(exception)
25         print(image_file)
26         break
27
28     image = cv2.resize(image, (img_size, img_size))
29
30     images_array = np.copy(images_array[index - len(dataset) // 2])
31
32 if images_array is None:
33     images_array = image.reshape(1, *image.shape)
34 else:
35     images_array = np.row_stack([
36         images_array, image.reshape(1, *image.shape)
37     ])
38
39 row = dataset.iloc[index]
40 array_ok.append([
41     row['ok']
42 ])
43
44 additional_X.append([
45     row['Женские'],
46     row['Мужские'],
47     row['Демисезонные'],
48     row['Зимние'],
49     row['Летние'],
50     row['Вечерние образы'],
51     row['Деловой стиль'],
52     row['Повседневные образы'],
53     row['Спорт']
54 ])
55
56 index(['Unnamed: 0', 'Женские', 'Мужские', 'Демисезонные', 'Зимние', 'Летние',
57     'Вечерние образы', 'Деловой стиль', 'Повседневные образы', 'Спорт',
58     'filepath', 'ok'],
59     dtype='object')
60
61 # Проверяем длину всех массивов
62 len(images_array), len(additional_X), len(array_ok)
63
64 (1904, 1904, 1904)
    
```

Рисунок 7 – Загружаем новую таблицу и разбиваем ее данные на массивы

## 2.2 Модель

```

[ ] 1 # разбиение данных на train/test
2
3 from sklearn.model_selection import train_test_split
4 from sklearn.model_selection import cross_val_score
5
6 images_array_train, images_array_test, additional_X_train, additional_X_test, array_ok_train, array_ok_test = train_test_split(
7     images_array, np.array(additional_X).reshape(
8         len(images_array), -1
9     ), np.array(array_ok).reshape(-1, 1),
10     random_state=1, # зафиксируем генератор случайных чисел для воспроизводимости
11     test_size=0.1, # 10% тестовое множество
12     stratify=array_ok # сохранять пропорции классов в разбиениях
13     # также по умолчанию этот метод перемешивает данные
14 )
15
16 print('images_array train shape:', images_array_train.shape, 'images_array test shape:', images_array_test.shape)
17 print('additional_X train shape:', additional_X_train.shape, 'additional_X test shape:', additional_X_test.shape)
18 print('array_ok train shape:', array_ok_train.shape, 'array_ok test shape:', array_ok_test.shape)
19
20 images_array train shape: (1713, 100, 100, 3) images_array test shape: (191, 100, 100, 3)
21 additional_X train shape: (1713, 9) additional_X test shape: (191, 9)
22 array_ok train shape: (1713, 1) array_ok test shape: (191, 1)
23
24 2.2.1.
25
26 [ ] 1 # Смотрим размер 2-х тренировочных множеств
27
28 3 images_array_train[0].shape, additional_X_train[0].shape
29
30 ((100, 100, 3), (9,))
    
```

Рисунок 8 – Разбиваем данные на 2 вида(тренировочные и тестовые)

```

1 # Создаем сетку для нейронной сети
2 import keras
3
4 keras.backend.clear_session()
5
6 def create_model(input_shape=(img_size, img_size, 3), input_vector_shape=(9, )):
7     image_input = keras.layers.Input(shape=input_shape)
8     vector_input = keras.layers.Input(shape=input_vector_shape)
9
10    layer = keras.layers.Conv2D(filters=8, kernel_size=(3, 3), padding='same', activation='relu')(
11        image_input
12    )
13    layer = keras.layers.MaxPool2D((3, 3))(layer)
14    layer = keras.layers.Dropout(0.1)(layer)
15    layer = keras.layers.Conv2D(filters=8, kernel_size=(3, 3), padding='same', activation='relu')(
16        layer
17    )
18    layer = keras.layers.MaxPool2D((2, 2))(layer)
19    layer = keras.layers.Dropout(0.1)(layer)
20    layer = keras.layers.Flatten()(layer)
21
22    layer = keras.layers.concatenate()([layer, vector_input])
23    layer = keras.layers.Dropout(0.1)(layer)
24    layer = keras.layers.Dense(18, activation='relu')(layer)
25    output = keras.layers.Dense(1, activation='sigmoid')(layer)
26
27    model = keras.Model([image_input, vector_input], output)
28    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
29
30    return model
31
32 classifier = create_model()
33 classifier.summary()
    
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 100, 100, 3)]	0	
conv2d (Conv2D)	(None, 100, 100, 8)	224	input_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 33, 33, 8)	0	conv2d[0][0]
dropout (Dropout)	(None, 33, 33, 8)	0	max_pooling2d[0][0]
conv2d_1 (Conv2D)	(None, 33, 33, 8)	584	dropout[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 8)	0	conv2d_1[0][0]
dropout_1 (Dropout)	(None, 16, 16, 8)	0	max_pooling2d_1[0][0]
flatten (Flatten)	(None, 2048)	0	dropout_1[0][0]
input_2 (InputLayer)	[(None, 9)]	0	
concatenate (Concatenate)	(None, 2057)	0	flatten[0][0] input_2[0][0]
dropout_2 (Dropout)	(None, 2057)	0	concatenate[0][0]
dense (Dense)	(None, 18)	37044	dropout_2[0][0]
dense_1 (Dense)	(None, 1)	19	dense[0][0]

```

1 # Смотрим размер тренировочного множества array_ok,
2 array_ok_train.shape, array_ok_train[0]
((1713, 1), array([0]))
    
```

Рисунок 9 – Создание сетки для обучения сети

```

1 # Обучаем сеть
2 validation = None # int(len(images_array_train) * 0.2)
3
4 classifier.fit(
5     x=[images_array_train[validation:] / 255., additional_X_train[validation:]],
6     y=array_ok_train[validation:],
7     epochs=200,
8     batch_size=1,
9     validation_data=(
10        images_array_test / 255., additional_X_test,
11        array_ok_test
12    ),
13    callbacks=[
14        keras.callbacks.ModelCheckpoint(
15            'model.hs', save_best_only=True, monitor='val_acc'
16        )
17    ]
18 );
    
```

```

[ ] 1 classifier = keras.models.load_model('model.hs')
2.2.2. Результат и оценка качества модели.
[ ] 1 # Вывод лучшего результата
2 classifier.evaluate([images_array_test / 255., additional_X_test], array_ok_test);
6/6 [=====] - 0s 50ms/step - loss: 0.6320 - acc: 0.7120
    
```

```

Epoch 1/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.7013 - acc: 0.4991 - val_loss: 0.6917 - val_acc: 0.5236
Epoch 2/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.6912 - acc: 0.4939 - val_loss: 0.6782 - val_acc: 0.5707
Epoch 3/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.6800 - acc: 0.5610 - val_loss: 0.6627 - val_acc: 0.6021
Epoch 4/200
1713/1713 [=====] - 15s 9ms/step - loss: 0.6723 - acc: 0.5657 - val_loss: 0.6557 - val_acc: 0.5969
Epoch 5/200
1713/1713 [=====] - 16s 9ms/step - loss: 0.6682 - acc: 0.5663 - val_loss: 0.6527 - val_acc: 0.5969
Epoch 6/200
1713/1713 [=====] - 15s 9ms/step - loss: 0.6642 - acc: 0.5668 - val_loss: 0.6529 - val_acc: 0.6073
Epoch 7/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.6652 - acc: 0.5651 - val_loss: 0.6444 - val_acc: 0.6178
Epoch 8/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.6602 - acc: 0.5692 - val_loss: 0.6316 - val_acc: 0.6492
Epoch 9/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.6440 - acc: 0.6089 - val_loss: 0.6320 - val_acc: 0.7120
Epoch 10/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.6296 - acc: 0.6328 - val_loss: 0.6260 - val_acc: 0.6702
Epoch 11/200
1713/1713 [=====] - 14s 8ms/step - loss: 0.6137 - acc: 0.6626 - val_loss: 0.6287 - val_acc: 0.6440
Epoch 12/200
1713/1713 [=====] - 17s 10ms/step - loss: 0.5930 - acc: 0.6824 - val_loss: 0.6158 - val_acc: 0.6649
Epoch 13/200
    
```

Рисунок 10 - Обучение сети

### Список литературы и иных ресурсов:

1. Бастиан Шарден, Лука Массарон, Альберто Боскетти «Крупномасштабное машинное обучение вместе с Python», 2018
2. <https://github.com/>
3. <https://stackoverflow.com/>
4. <https://www.youtube.com/>