

Распознавание эмоций для улучшения опыта посещения музеев

Команда “Улыбка Тьюринга”:

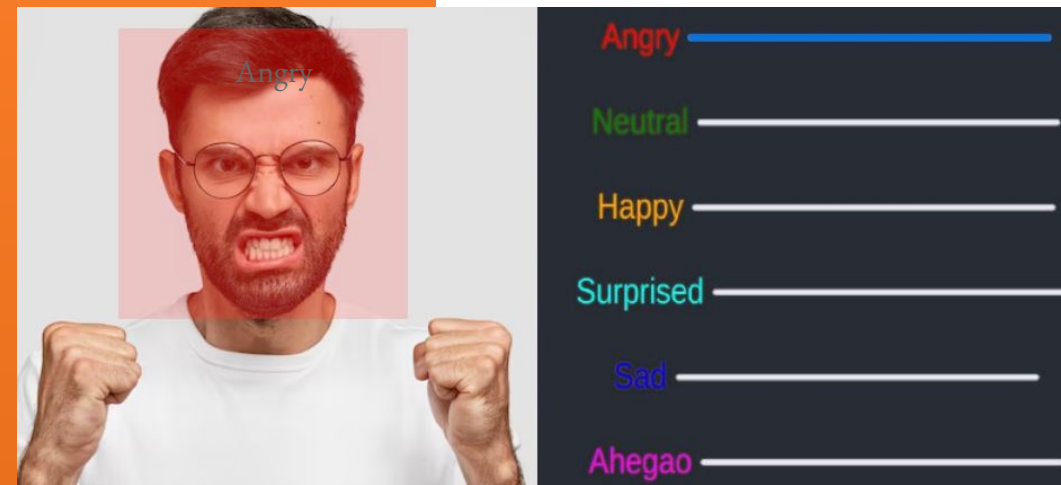
Кубракова Екатерина Адександровна

Земскова Мария Викторовна

Анисимов Юрий Сергеевич

Ченчак Михаил Андреевич

<https://github.com/aniyur/skillfactory-ds-hackaton-23>



Решаемая задача

Задача проекта — улучшить опыт от посещения музеев, городов или новых мест с помощью информационных технологий.

Цель проекта — сделать посещение людьми музеев, городов или новых мест интересным, информативным и комфортным.

Для выполнения задачи реализуем систему распознавания эмоций, которую можно будет использовать для автоматического определения получаемых людьми эмоций от просмотра музейных экспонатов.

Музеи смогут:

- предлагать людям для посещения на выбор места, вызывающие интересующие их эмоции;
- подбирать наиболее интересующие людей экспонаты и т.п.

Данные



Входные данные

Нами использовался датасет из открытого источника (Kaggle):

<https://www.kaggle.com/datasets/sujaykapadnis/emotion-recognition-dataset/data>

Датасет содержит фотографии людей, лица которых выражают 6 типов эмоций:

1. счастье
2. страх
3. удивление
4. радость
5. грусть
6. экстаз

Анализ данных

Angry



Happy



Sad



Neutral



Angry



Happy



Sad



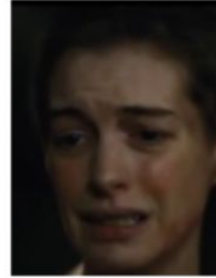
Neutral



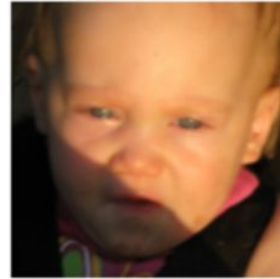
Sad



Sad



Sad



Happy



Sad



Neutral



Angry



Ahegao



Neutral

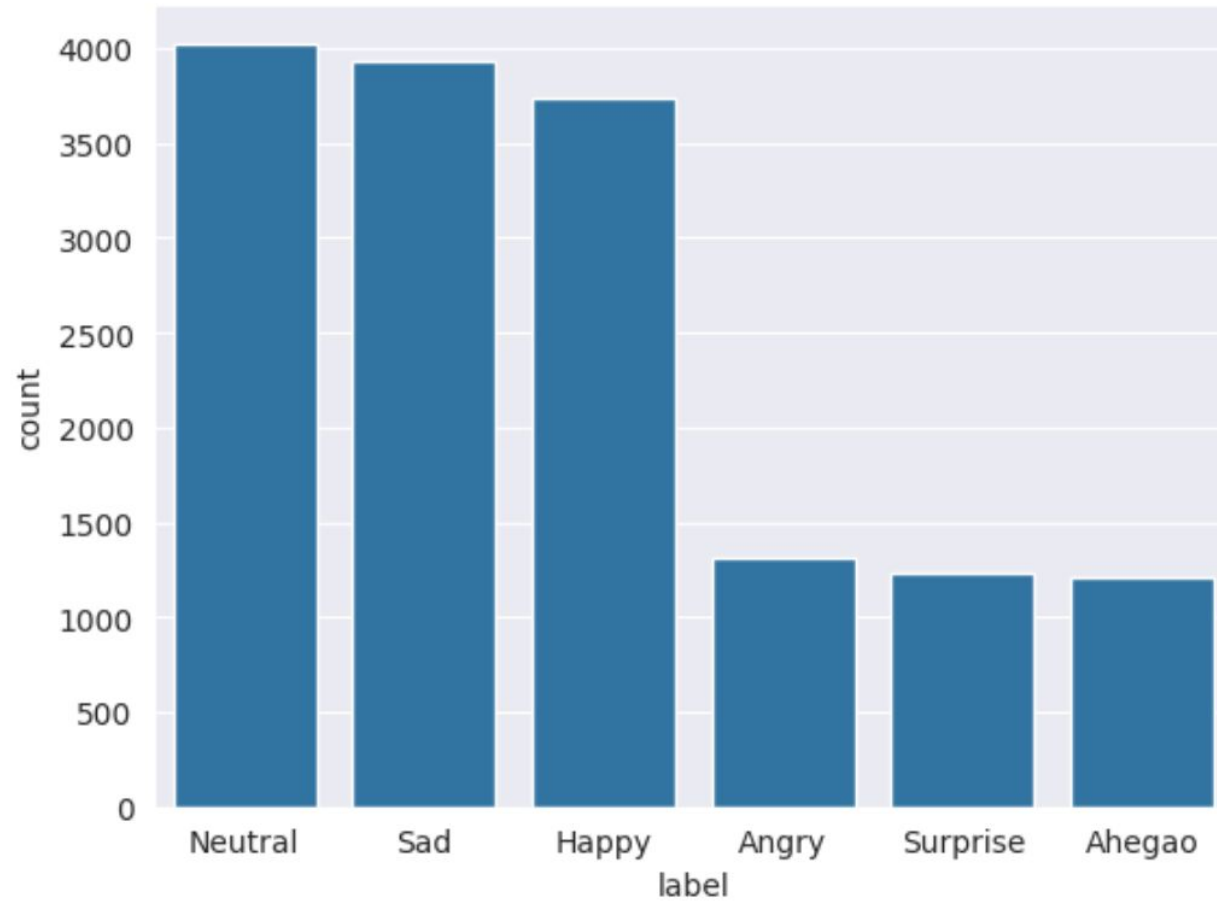


Neutral



Обнаружено, что картинки имеют разный размер и пропорции, отличаются по яркости, контрастности, наклону, повороту головы и т.п.

Анализ данных



Выявлен дисбаланс классов

Подготовка данных

В ходе подготовки данных, нами были предприняты следующие шаги:

1. Определены значения для нормализации (по тренировочным данным) и нормализованы значения интенсивностей цветов (torch transforms)
2. Картинки приведены к единому формату 224x224 пкс.
3. К тренировочным данным применены различные аугментации:
 - поворот
 - изменение масштаба
 - сдвиг
 - наклон
 - изменение яркости
 - изменение контрастности
 - изменение насыщенности
 - изменение цвета

Набор разбили на 80% тренировочных примеров и 20% валидационных (12 тыс. к 3 тыс.).

Пример аугментации

Sad



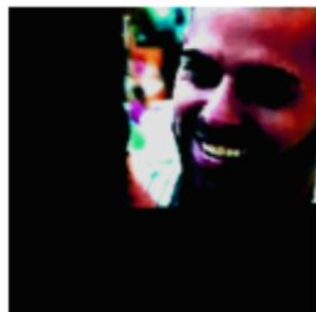
Happy



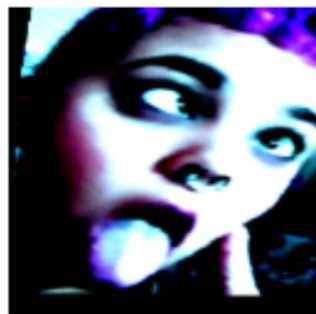
Sad



Happy



Ahegao



Sad



Sad



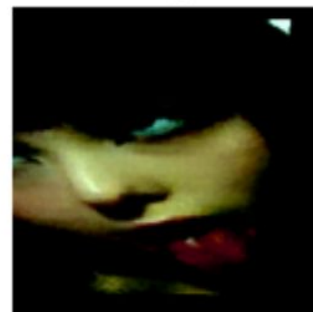
Neutral



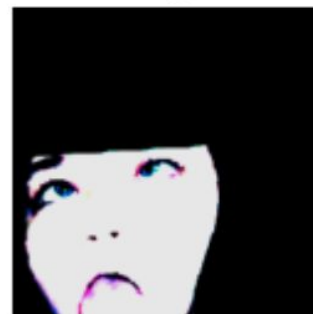
Neutral



Ahegao



Ahegao



Angry



Happy



Happy



Sad



Подбор модели



Базовая модель

За базовую модель
взяли обычную
свёрточную сеть (Pytorch)

```
lr = 0.001  
optimizer = torch.optim.Adam
```

```
Epoch 0 train loss: 1.5976, val loss: 1.4969, val acc: 0.3181  
Epoch 1 train loss: 1.3818, val loss: 1.2808, val acc: 0.4834  
Epoch 2 train loss: 1.1481, val loss: 1.1483, val acc: 0.5280  
Epoch 3 train loss: 0.9962, val loss: 0.9960, val acc: 0.5988  
Epoch 4 train loss: 0.9154, val loss: 0.9445, val acc: 0.6307
```

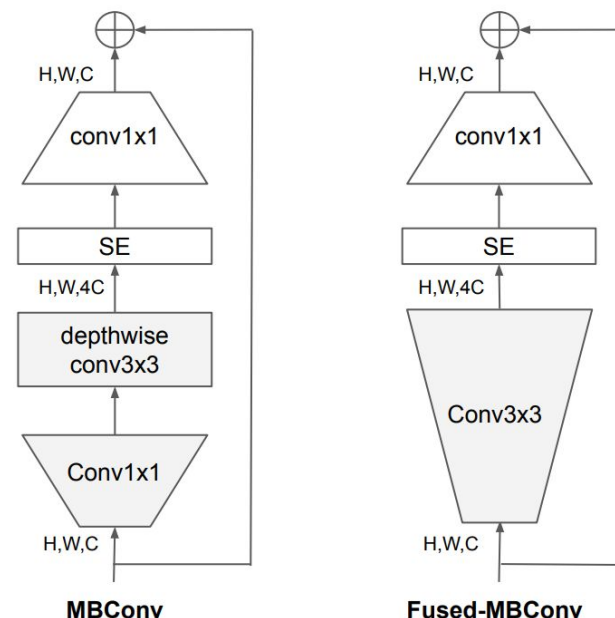
60%-точность, быстрое переобучение

```
class Model(nn.Module):  
    def __init__(self):  
        super().__init__()  
        self.layers = nn.Sequential(  
            nn.Conv2d(3, 32, kernel_size=3, padding=1),  
            nn.ReLU(),  
            nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),  
            nn.ReLU(),  
            nn.MaxPool2d(2,2),  
  
            nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1),  
            nn.ReLU(),  
            nn.Conv2d(128, 128, kernel_size=3, stride=1, padding=1),  
            nn.ReLU(),  
            nn.MaxPool2d(2,2),  
  
            nn.Flatten(),  
            nn.Linear(8192, 256),  
            nn.ReLU(),  
            nn.Linear(256,6),  
        )
```

EfficientNetV2

Средний вариант:
50M+ параметров

Предобученные параметры:
ImageNet1K_V1 (Pytorch)



<https://arxiv.org/pdf/2104.00298.pdf>

Подбор модели

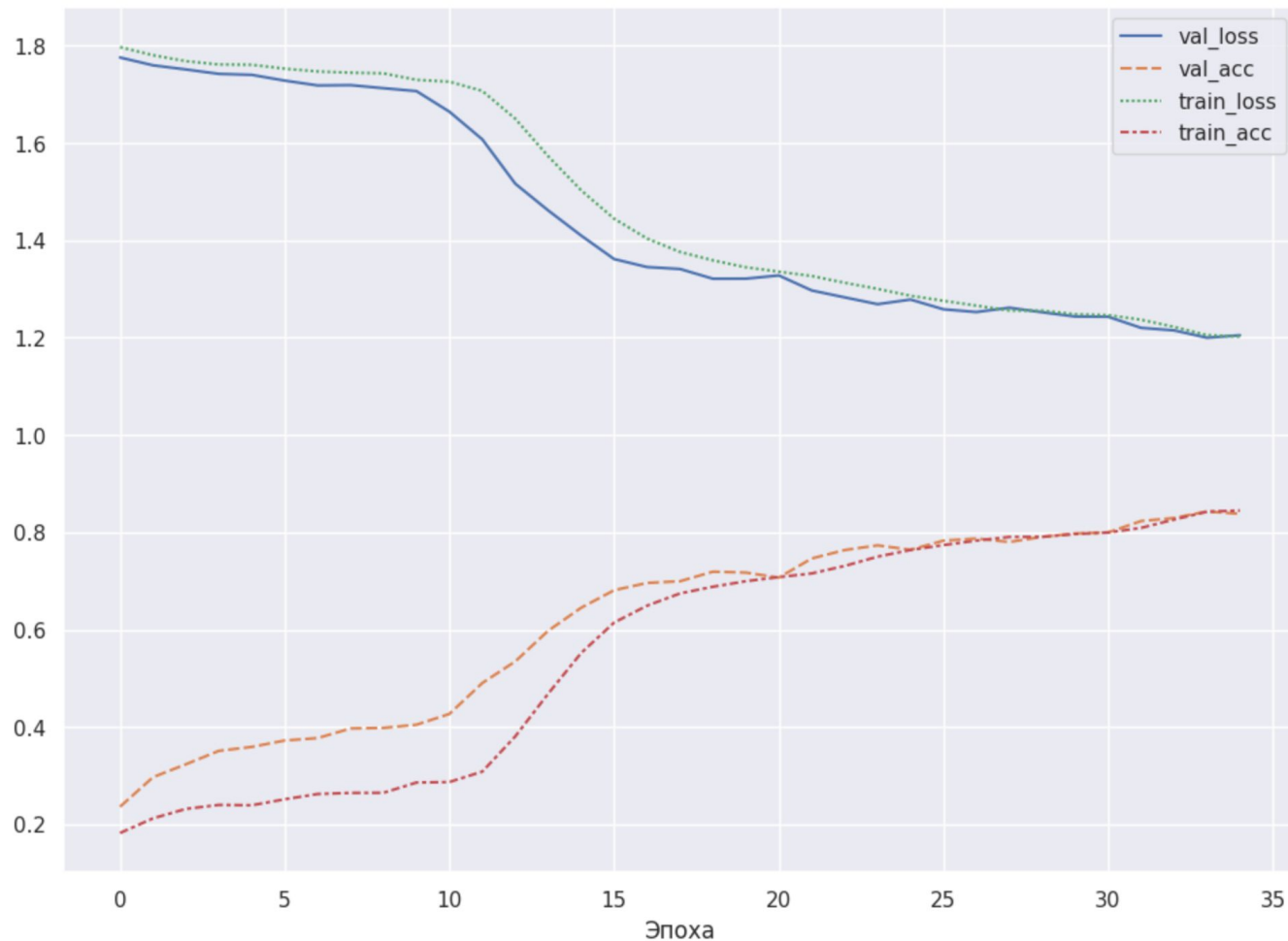
Для предотвращения переобучения также применён **Dropout 0,98**

```
(classifier): Sequential(
  (0): Dropout(p=0.98, inplace=True)
  (1): Linear(in_features=1280, out_features=6, bias=True)
  (2): Softmax(dim=1)
```

Рабочая модель получена за 30 эпох обучения (обучение остановлено до того, как показатели валидации начали расходиться с показателями тренировки (i.e. “early stopping”)):

```
Epoch 30 train loss: 1.2076, val loss: 1.1994, train acc: 0.8405, val acc: 0.8434
```

Обучение



10 эпох
оптимизировали
веса выходного
слоя, затем всей
модели

Epoch 30 train loss: 1.2076, val loss: 1.1994, train acc: 0.8405, val acc: 0.8434

Запуск приложения

Для использования нашей модели доработано веб-приложение “facial-emotion-recognition”, определяющее границы лица и эмоции.

- добавлено использование определяемых координат границ лица для вырезания соответствующей части изображения, передаваемого веб-камерой, для последующей отправки для инференса эмоций в бэковую (не браузерную) часть приложения
- реализовано использование нашей модели в бэковой части (наша нормализация, наша модель)

исходное приложение:

<https://github.com/victor369basu/facial-emotion-recognition>

наша доработанная версия (форк), там же ноутбук и описание:

<https://github.com/ds-journey/emotion-recognition>

Пример работы приложения



**Спасибо
за внимание**

