



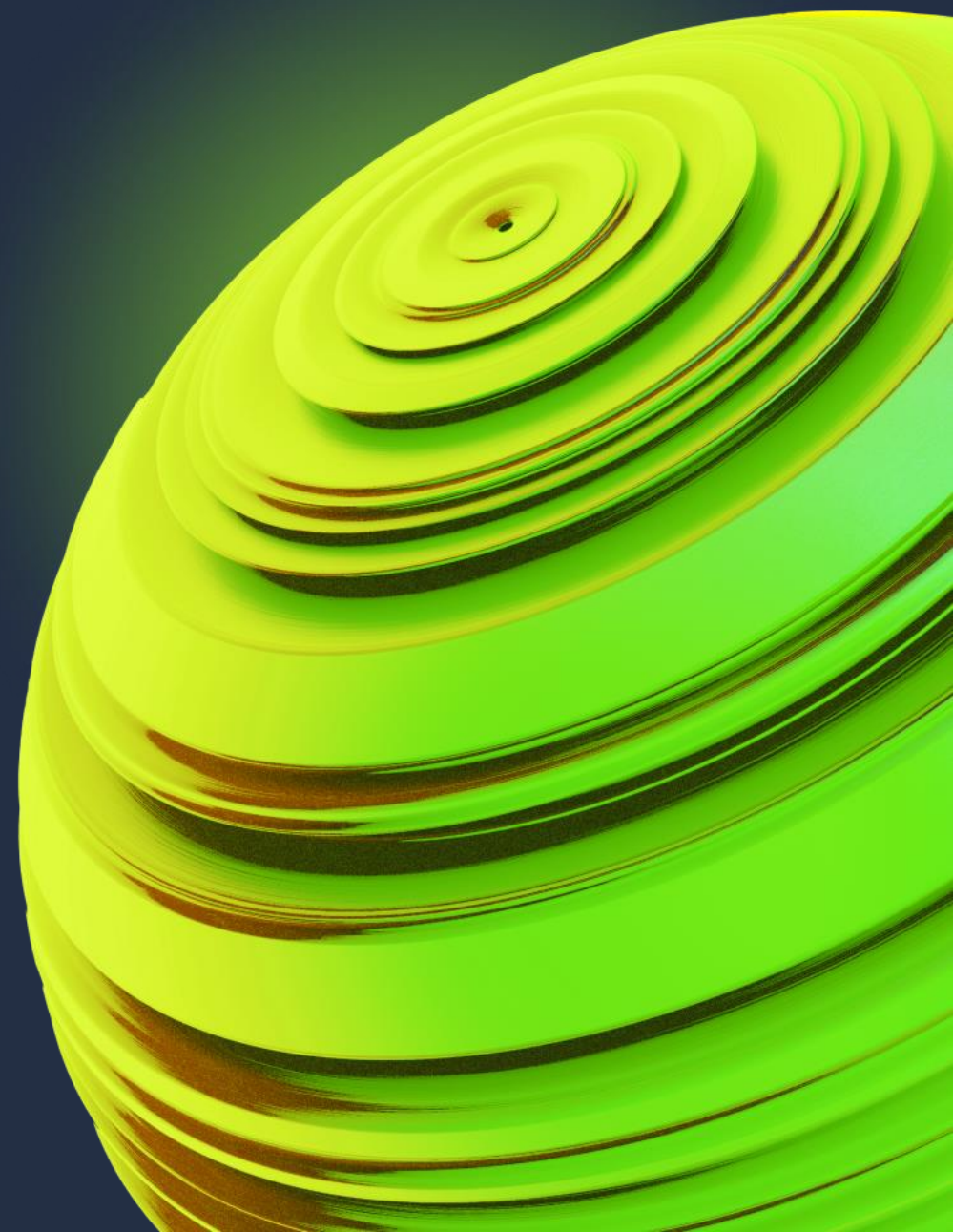
ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС



УНИВЕРСИТЕТ
ИННОПОЛИС

Проект 9. Распознавание эмоций на видео

Итоговое задание для курса “Профессия ML-инженер”



Актуальность работы



Распознавание эмоций на видео может использоваться для:

- Персонализированной рекламы.
- Работа-психолога
- Работа-консультанта
- Получение обратной связи от клиентов без опросов
- Анализа реакции аудитории на фильм/сериал и т.д.

Цели и задачи



Цель:

Обучить модели для распознавания эмоции на видео

Задачи:

- Реализация модуля поиска лица на видео
- Обучение модели распознавания не менее 5 эмоций
- Реализация интерфейса, позволяющий считывать эмоции человека с видео или веб-камеры, с выводом результата и вероятностью эмоций

Применяемые технологии



- [Pytorch](#) – построение моделей
- [Ultralytics](#) - модель Yolo11n
- [Moviepy](#) - сохранение полученного видео в формате mp4
- [OpenCV](#)- модуль детекции лица([haarcascade_frontalface_default.xml](#)), работа с кадрами и датасетом
- [Gradio](#) - создание веб-интерфейса
- [ONNX](#), [ONNX Runtime](#) - конвертирование моделей в формат .onnx и ускорение вывода модели
- [Hugging Face](#) - деплой веб-интерфейса

Этапы работы над проектом



1. Анализ существующих решений, подбор и загрузка релевантного датасета
2. Выбор моделей для обучения
3. Обучение моделей
4. Оценка качества моделей
5. Экспорт моделей в формат .onnx
6. Создание веб-интерфеса для работы с моделями
7. Деплой приложение на [HuggingFace](https://huggingface.co)

Анализ существующих решений, подбор и загрузка релевантного датасета



Существует множество различных моделей и датасетов для распознавания эмоций на видео.

Для данной работы был выбран датасет [FER2013](#). Он состоит из 35887 черно-белых изображений размером 48x48



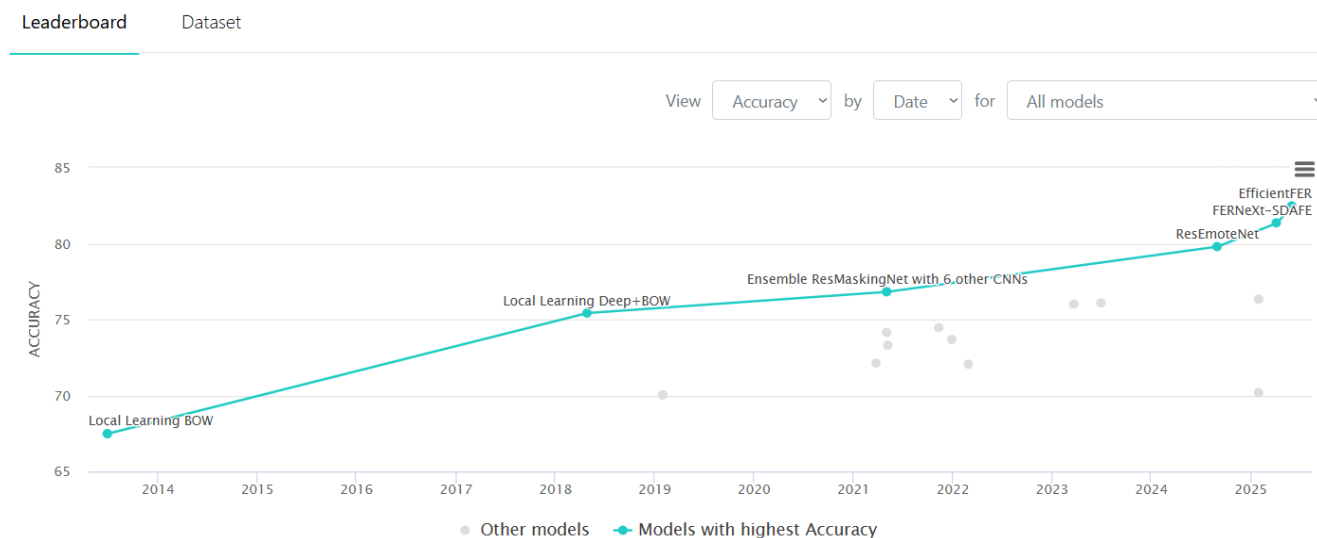
Выбор моделей для обучения



На сайте [Paperswithcode](https://paperswithcode.com) представлены модели с лучшими тестовыми показателями на выбранном датасете.

Для данной работы было решено обучить модели EfficientNet, ResNet18. В качестве эксперимента было также обучена модель Yolo11n

Facial Expression Recognition (FER) on FER2013



Подготовка датасета 1



Датасет разбит только на тестовую(7178) и обучающую(28709) выборки.

Датасет для тестирования составляет примерно 20% от общего количества изображений. Возьмём 20% от обучающего датасета и создадим из неё валидационную выборку(5748).

Подготовка датасета 2



Аугментация датасета с помощью torchvision.transforms.

```
train_transform = transforms.Compose(  
    [transforms.Resize(size=(224, 224)),  
     v2.RandomHorizontalFlip(p=0.5),  
     v2.RandomVerticalFlip(p=0.4),  
     v2.RandomRotation(degrees=(0, 90)),  
     v2.ColorJitter(brightness=random.uniform(1,0.5), contrast=random.random()),  
     transforms.ToTensor()]  
)  
  
val_transform = transforms.Compose(  
    [transforms.Resize(size=(224, 224)),  
     transforms.ToTensor()]  
)
```

Визуализация датасета

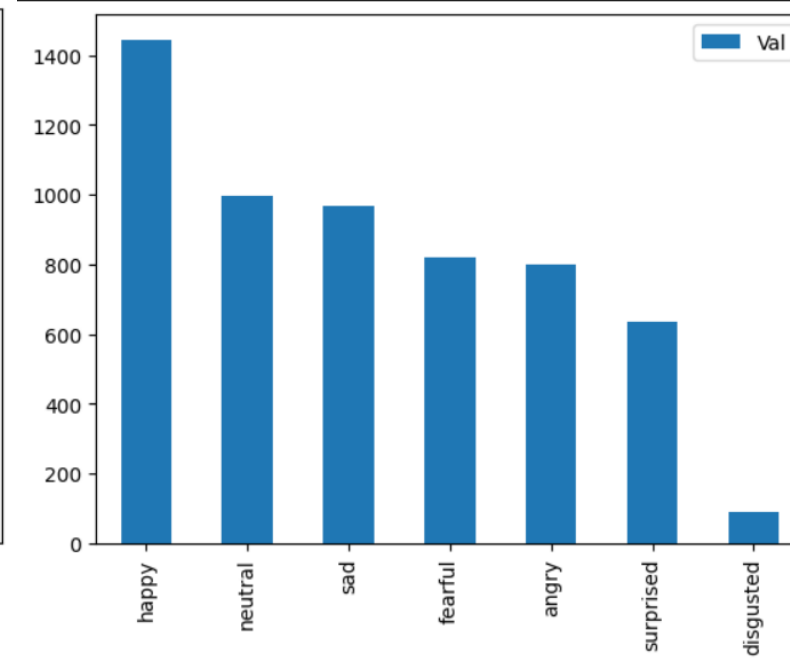
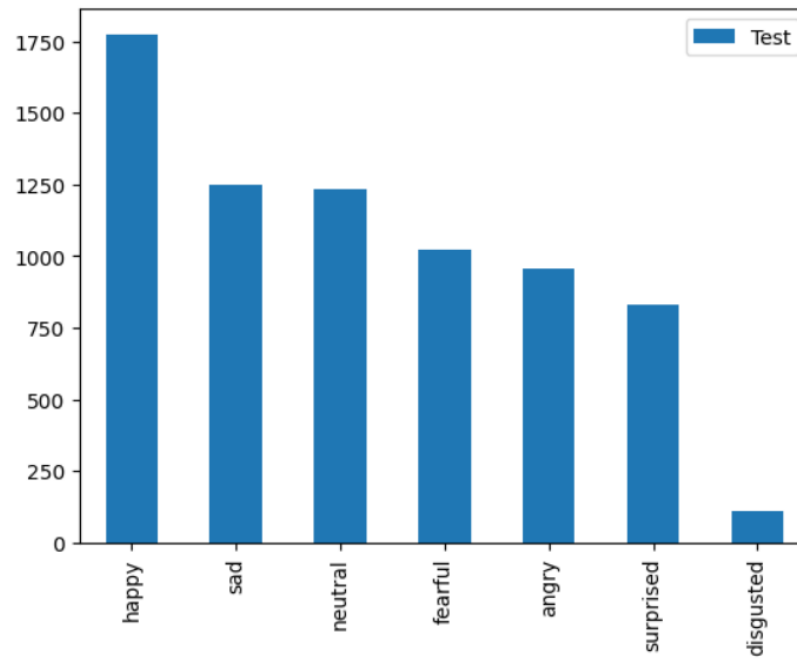
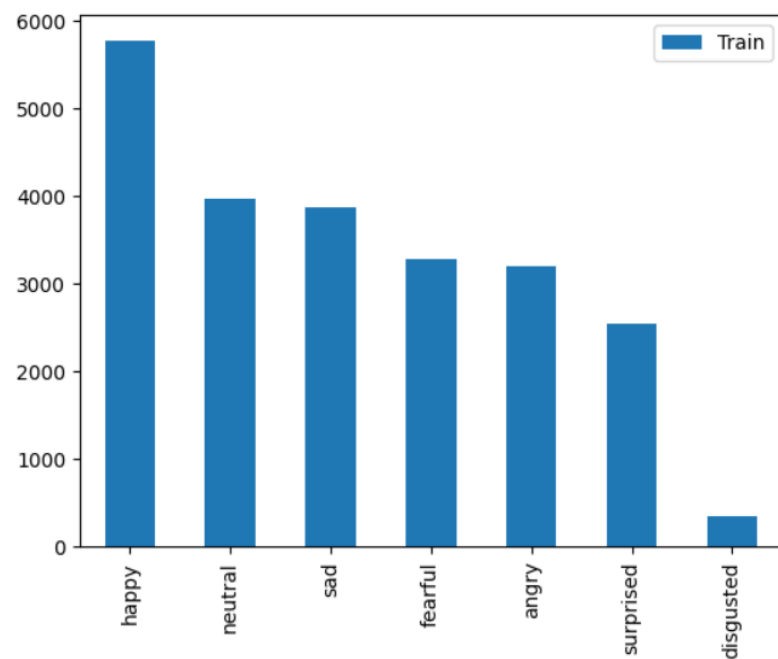
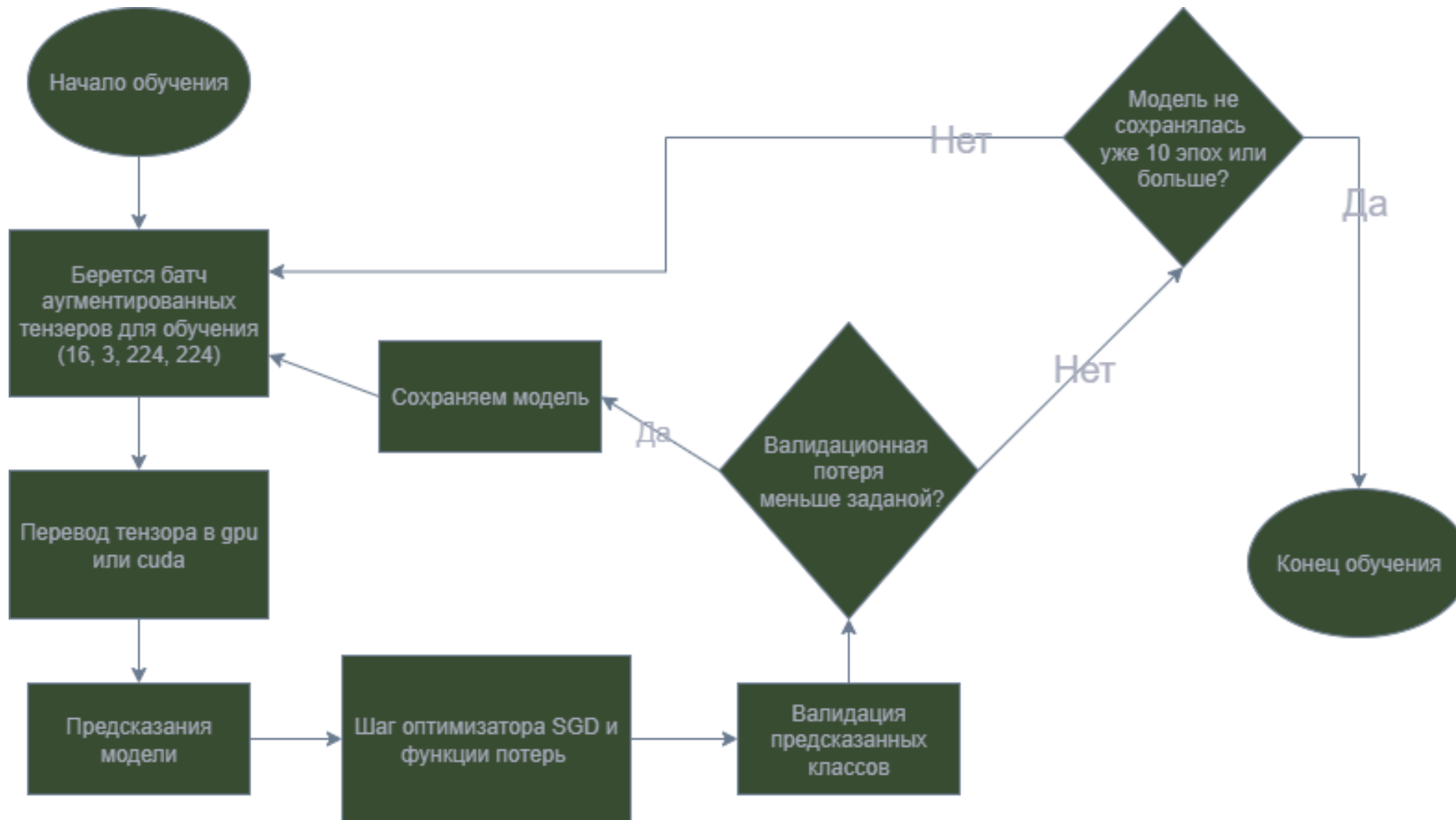


Диаграмма пайплайна обучения(EfficientNet, ResNet18)



Оценка качества моделей(EfficientNet)



Confusion Matrix

True Labels	angry	525	28	102	33	115	135	20
	disgusted	19	61	11	4	4	10	2
	fearful	105	12	442	44	124	213	84
	happy	19	1	16	1563	91	36	48
	neutral	47	2	51	94	852	174	13
	sad	93	11	112	60	257	700	14
	surprised	20	3	74	30	34	10	660
		angry	disgusted	fearful	happy	neutral	sad	surprised

Classification Report:

	precision	recall	f1-score	support
angry	0.63	0.55	0.59	958
disgusted	0.52	0.55	0.53	111
fearful	0.55	0.43	0.48	1024
happy	0.86	0.88	0.87	1774
neutral	0.58	0.69	0.63	1233
sad	0.55	0.56	0.55	1247
surprised	0.78	0.79	0.79	831
accuracy			0.67	7178
macro avg	0.64	0.64	0.63	7178
weighted avg	0.67	0.67	0.67	7178

Оценка качества моделей(ResNet18)



Confusion Matrix

True Labels	angry	disgusted	fearful	happy	neutral	sad	surprised
angry	573	9	78	32	116	125	25
disgusted	59	30	5	5	1	11	0
fearful	187	8	339	42	126	213	109
happy	71	1	15	1537	78	28	44
neutral	94	2	70	87	774	175	31
sad	187	4	112	75	257	586	26
surprised	39	0	91	47	27	18	609
		Predicted Labels					

Classification Report:				
	precision	recall	f1-score	support
angry	0.47	0.60	0.53	958
disgusted	0.56	0.27	0.36	111
fearful	0.48	0.33	0.39	1024
happy	0.84	0.87	0.85	1774
neutral	0.56	0.63	0.59	1233
sad	0.51	0.47	0.49	1247
surprised	0.72	0.73	0.73	831
accuracy			0.62	7178
macro avg	0.59	0.56	0.56	7178
weighted avg	0.62	0.62	0.61	7178

Перевод модели в формат ONNX



```
example_inputs = (torch.randn(1, 3, 224, 224),)

torch.onnx.export(model,
                  example_inputs,
                  "EfficientNet.onnx",
                  export_params=True,
                  opset_version=11,
                  do_constant_folding=True,
                  input_names=['input'],
                  output_names=['output'])

print("Модель успешно экспортирована в формат ONNX!")
```

Формат ONNX
позволяет моделям
машинного
обучения работать
на разных
платформах

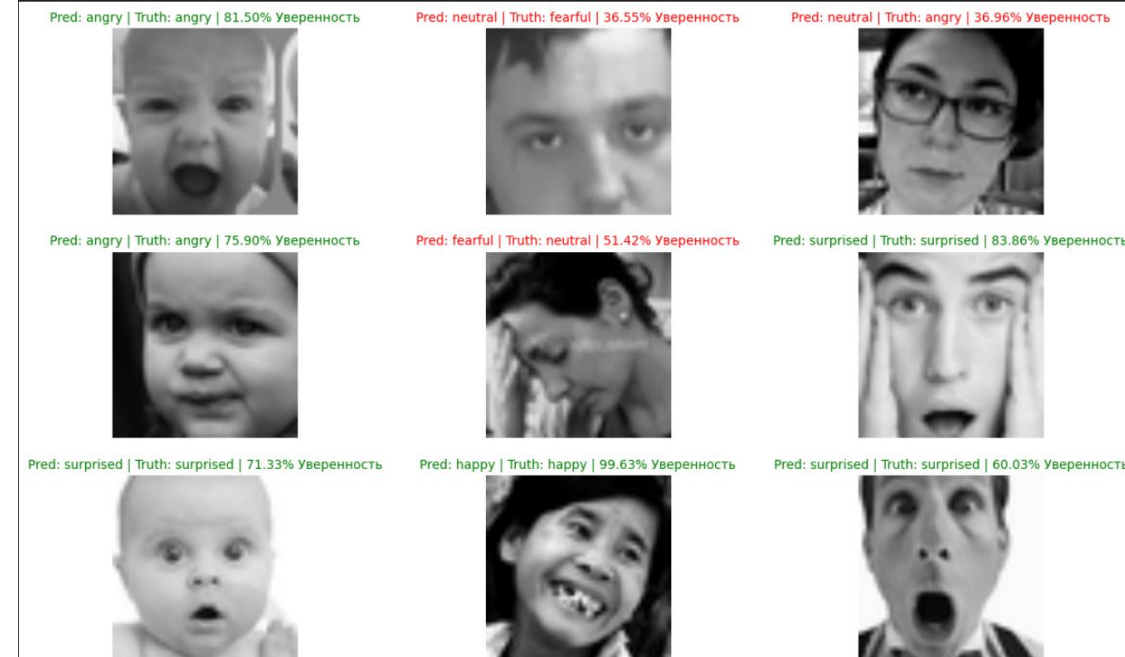
Диаграмма пайплайна инференса(EfficientNet, ResNet18)



Примеры распознавания(EfficientNet, ResNet18)



EfficientNet



ResNet18

Подготовка датасета 3



Для обучения модели Yolo11n сократим кол-во изображений в обучающей выборке, чтобы сократить время обучения.

Из каждого класса обучающего датасета возьмем не более 1500 фото. Кроме этого создадим новые папки test, train, validation. В них переместим изображения подпапку images, а в подпапку labels поместим файл с bounding box.

Подготовка датасета 4



```
train_path = images_path / "train"
validation_path = images_path / "validation"
test_path = images_path / "test"
class_names = ['angry', 'disgusted', 'fearful', 'happy', 'neutral', 'sad', 'surprised']

for path in [train_path, validation_path, test_path]:
    yolo_images = path / "images"
    yolo_labels = path / "labels"
    for dir_name in class_names:
        file_count = 0
        dir_path = path / dir_name
        for root, dir, files in os.walk(dir_path):
            for file in files:
                if file_count >= 1500 and path == train_path:
                    break
                if not os.path.exists(yolo_images):
                    os.makedirs(yolo_images)
                if not os.path.exists(yolo_labels):
                    os.makedirs(yolo_labels)
                file_count += 1
                shutil.copy(os.path.join(root, file), yolo_images / f"{dir_name}_{file}")
                label_file = open(yolo_labels / f"{dir_name}_{os.path.splitext(file)[0]}.txt", "w+")
                label_file.write(f"{class_names.index(dir_name)} 0.48 0.48 1.0 1.0")
                label_file.close()
```

Подготовка датасета 5



Подберем для модели гиперпараметры аугментации с помощью `model.tune()`

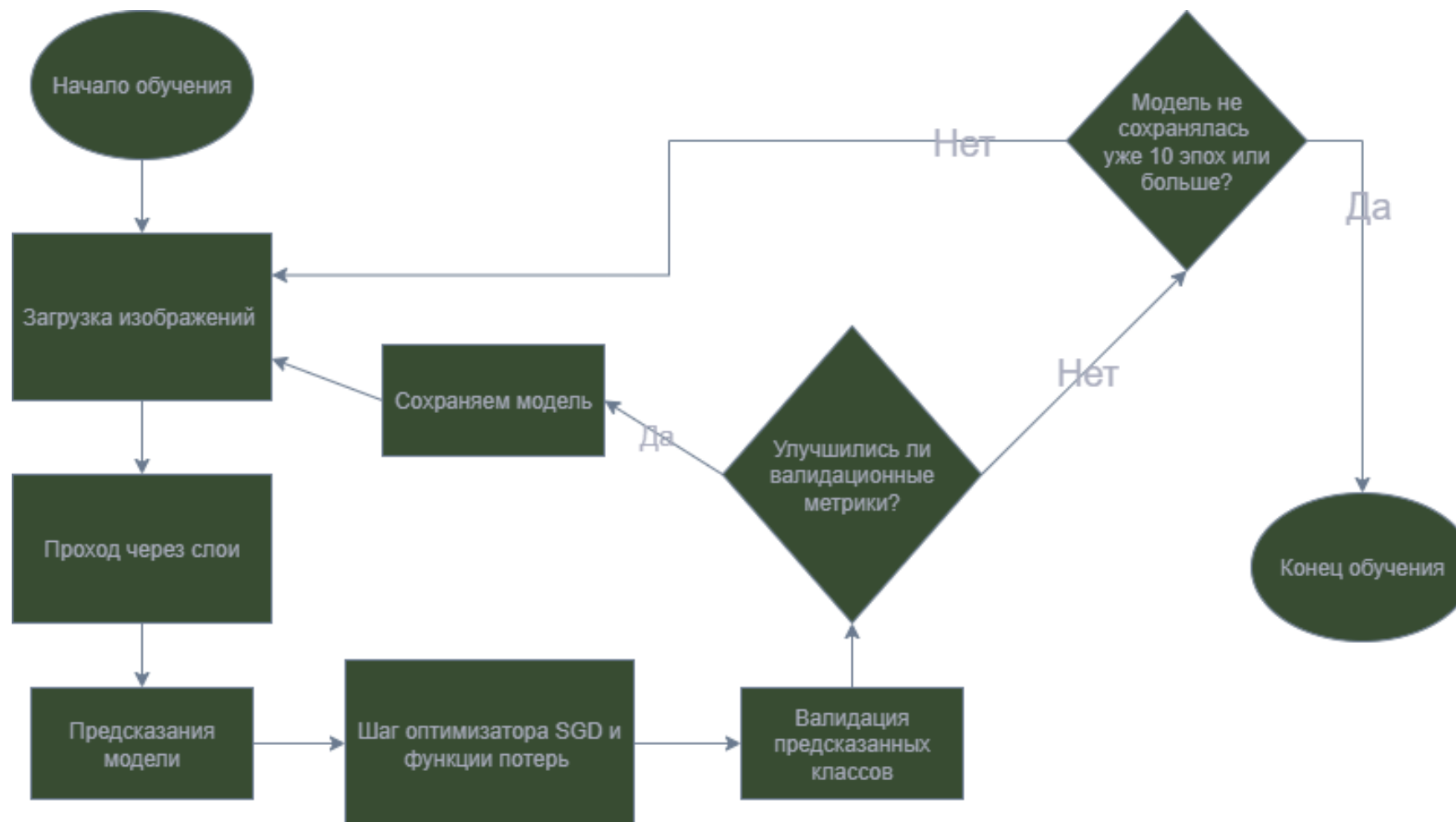
```
search_space = {
    "lr0": (1e-5, 1e-1),
    "degrees": (0.0, 90.0),
    "hsv_h" : (0.0, 0.1),
    "hsv_s" : (0.0, 0.9),
    "hsv_v" : (0.0, 0.9),
    "mosaic" : (0.0, 0.0),
    "fliplr": (0.0, 1.0),
    "mixup": (0.0, 0.0),
    "copy_paste": (0.0, 0.0),
    "scale" : (0.0, 0.9)
}

results = model.tune(
    data="dataset.yaml",
    epochs=3,
    iterations=20,
    optimizer="AdamW",
    space=search_space,
    plots=False,
    save=False,
    val=False,
)
```

```
lr0: 0.01
degrees: 0.0
hsv_h: 0.01525
hsv_s: 0.70708
hsv_v: 0.4063
mosaic: 0.0
fliplr: 0.5024
mixup: 0.0
copy_paste: 0.0
scale: 0.5
```

Полученные
гиперпараметры

Диаграмма пайплайна обучения(Yolo11n)

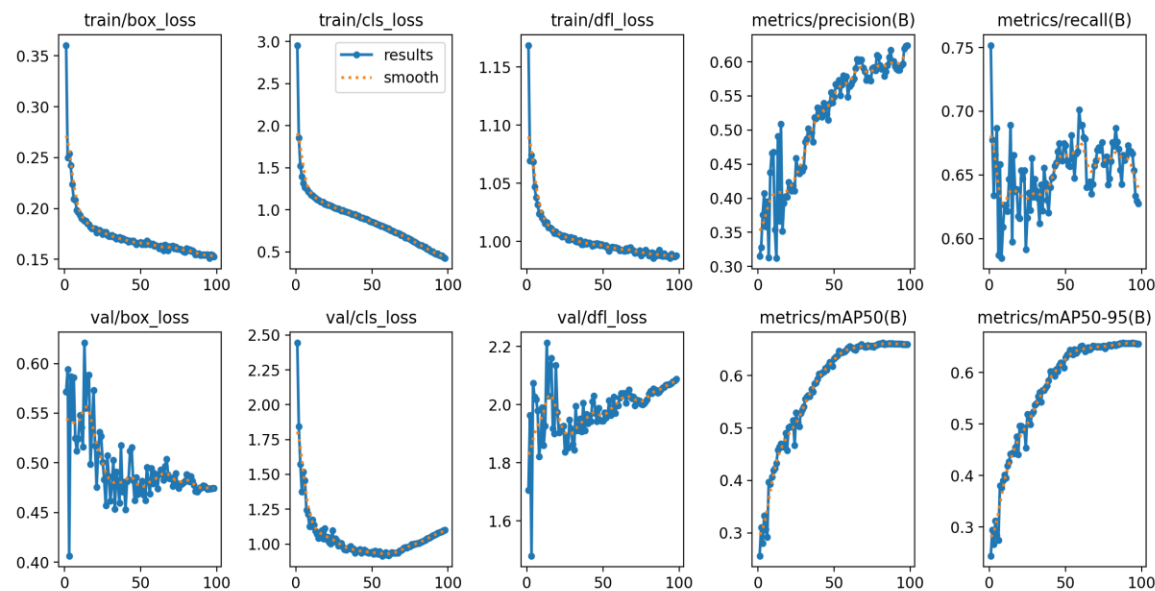


Оценка качества моделей(Yolo11n)

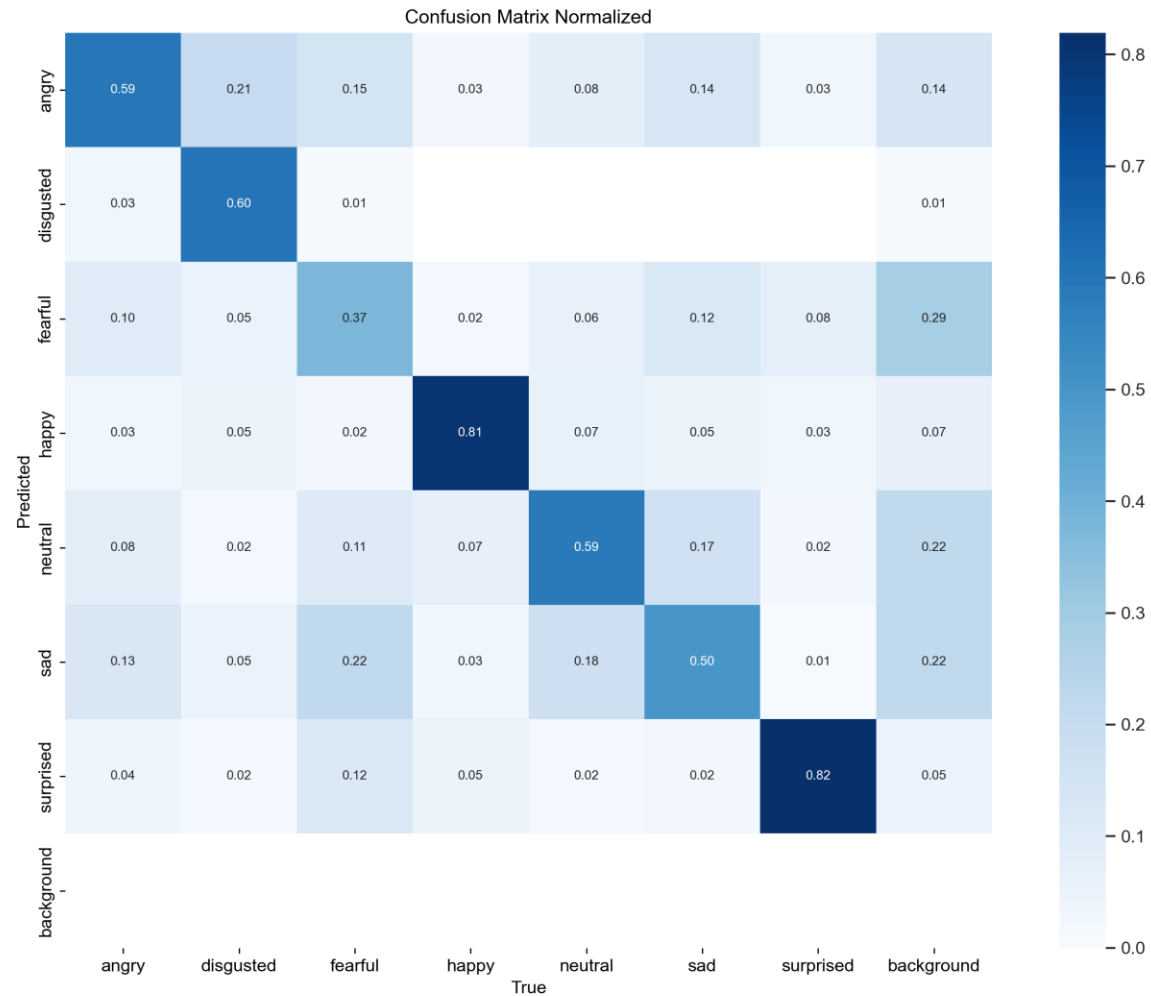


Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	7178	7178	0.584	0.658	0.654	0.651
angry	958	958	0.506	0.627	0.608	0.604
disgusted	111	111	0.544	0.613	0.614	0.612
fearful	1024	1024	0.448	0.493	0.469	0.463
happy	1774	1774	0.863	0.837	0.905	0.903
neutral	1233	1233	0.548	0.656	0.641	0.636
sad	1247	1247	0.491	0.542	0.518	0.514
surprised	831	831	0.691	0.836	0.825	0.823

Оценка на тестовой
выборке



Оценка качества моделей(Yolo11n)



Оценка на тестовой выборке

Перевод модели в формат ONNX



```
model.export(format="onnx")
```

```
Ultralytics 8.3.163 🚀 Python-3.11.13 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
```

```
PyTorch: starting from 'EmotionRec_Yolo11.pt' with input shape (1, 3, 640, 640) BCHW and output shape(s) (1, 11, 8400) (5.1 MB)
```

```
ONNX: starting export with onnx 1.18.0 opset 19...
```

```
ONNX: slimming with onnxslim 0.1.59...
```

```
ONNX: export success ✅ 2.6s, saved as 'EmotionRec_Yolo11.onnx' (10.1 MB)
```

```
Export complete (3.0s)
```

```
Results saved to /content
```

```
Predict: yolo predict task=detect model=EmotionRec_Yolo11.onnx imgsz=640
```

```
Validate: yolo val task=detect model=EmotionRec_Yolo11.onnx imgsz=640 data=dataset.yaml
```

```
Visualize: https://netron.app
```

```
'EmotionRec_Yolo11.onnx'
```

Формат ONNX позволяет моделям машинного обучения работать на разных платформах

Диаграмма пайплайна инференса(Yolo11n)



Примеры распознавания(Yolo11n)

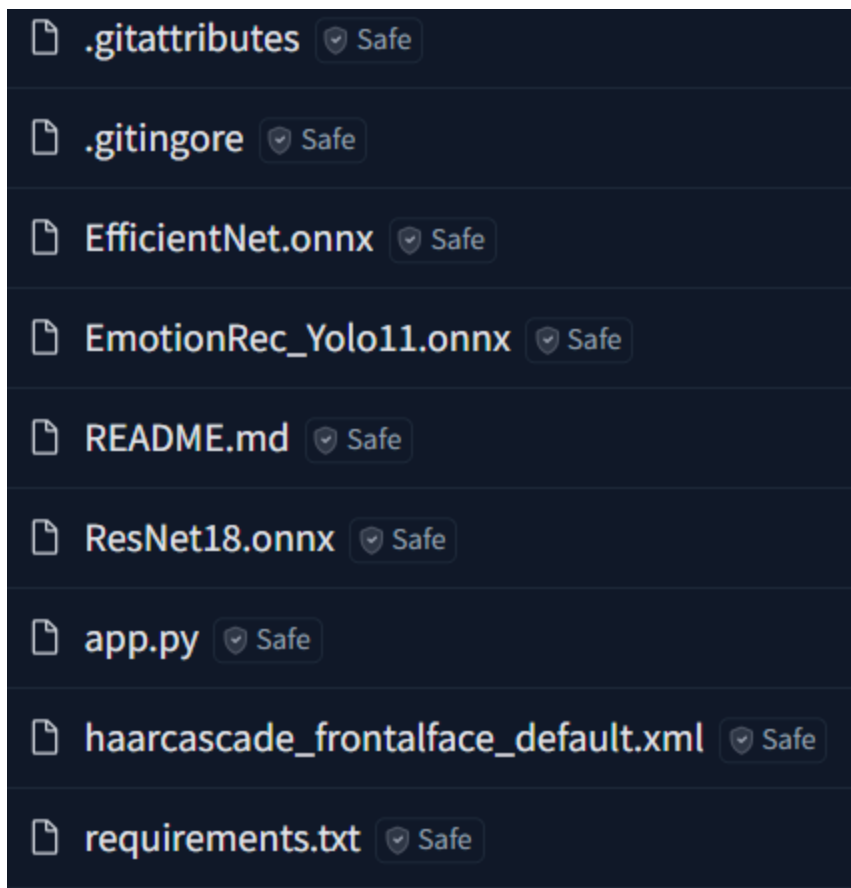


Настоящие классы



Предсказанные классы

Создание веб-интерфейса с помощью Gradio



```
gradio==5.35.0
gradio_client==1.10.4
moviepy==2.2.1
torch==2.7.1
ultralytics==8.3.40
opencv-python==4.11.0.86
onnx==1.18.0
onnxruntime==1.22.0
numpy==2.1.3
```

Файл requirements.txt

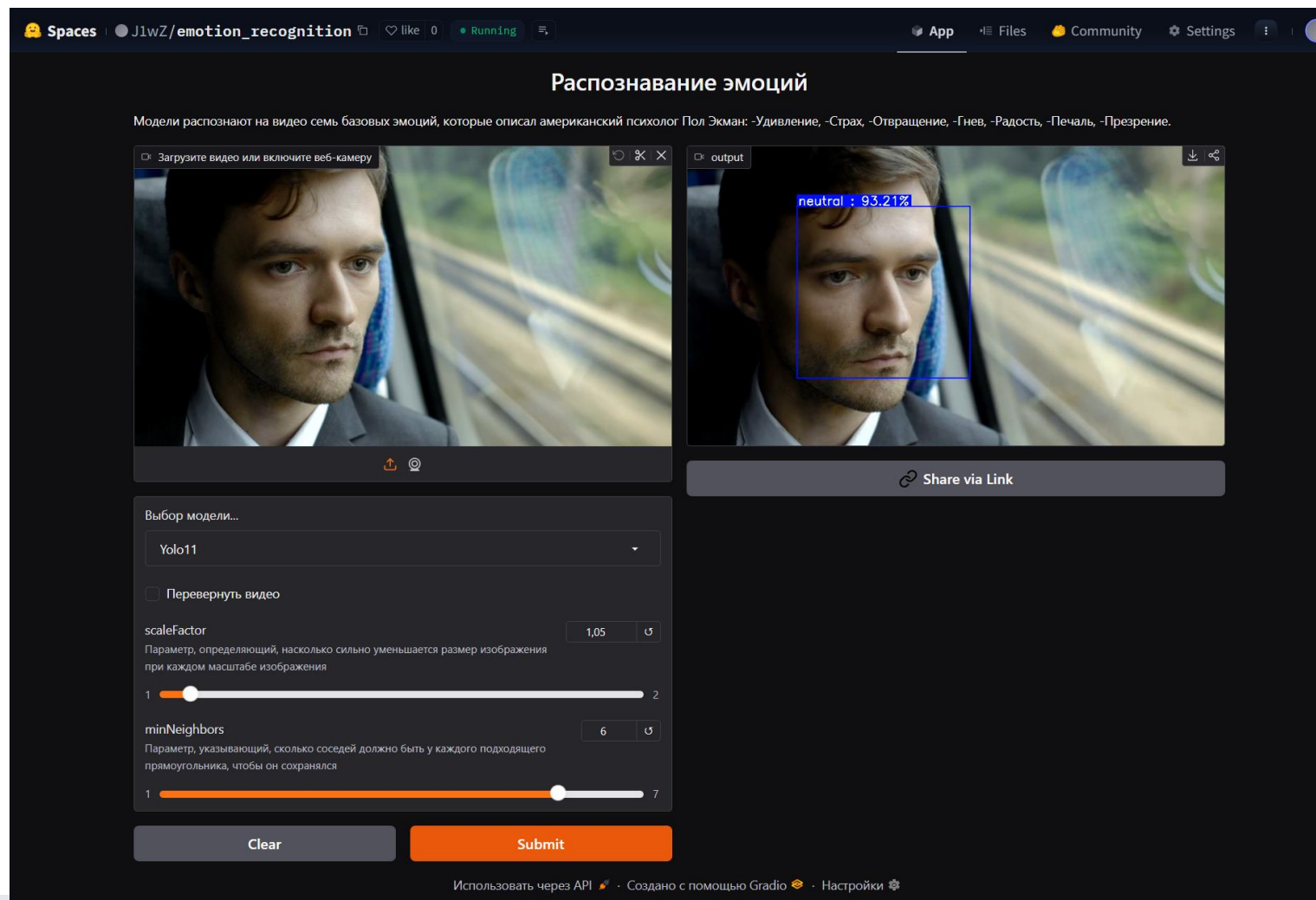
Создание веб-интерфейса с помощью Gradio



- Создается новое space в Hugging Face. Оно с помощью Git копируется на компьютер.
- В файле `app.py` пишется код интерфейса: функции для распознавания эмоций, нанесения результатов на кадр видео, сохранения нового видео и сам код интерфейса Gradio.
- Добавляется файл `.gitignore`, в котором указываются расширения файлов, что не нужно добавлять в репозиторий
- Добавляются модуль детекции лица `haarcascade_frontalface_default.xml` и модели в формате ONNX
- В файле `requirements.txt` указываются все необходимые для работы библиотеки

Деплой веб-интерфейса

С помощью Git загружаем все файлы на Hugging Face. [Ссылка](#)



Заключение



- С помощью haarcascade_frontalface_default.xml можно эффективно детектировать лица на кадрах видео
- Были обучены три модели распознающие 7 эмоций
- Лучшие метрики на тестовой выборке дала EfficientNet (Test loss: 0.915, Test accuracy: 0.669)
- Все модели дали относительно похожие результаты
- ONNXRuntime значительно уменьшил время инференса моделей
- Был создан веб-интерфейс с помощью Gradio, который выводит считанные эмоции и процент уверенности. Его можно [посмотреть на Hugging Face](#)
- Для просмотра кода обучения и видео примеров работы распознавания моделей перейти в [GitHub репозиторий](#)



ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС

Спасибо за внимание!

Контакты

 8 950 835 69 16

 Kali963caliborn@mail.ru



Сайт

<https://github.com/J1wZ>

<https://huggingface.co/J1wZ>

