

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Подразделение, ответственное за реализацию дополнительной профессиональной программы профессиональной переподготовки «Искусственный интеллект и машинное обучение» (далее-ДПП ПП) - Центр «Цифровая культура»

Руководитель ДПП ПП – к.т.н., доцент, зав. каф. «Программное обеспечение вычислительной техники и автоматизированных систем» В.В. Долгов

**ОТЧЕТ**

По практической подготовке при проведении практики в ООО «ИТР-ГРУПП»



Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Максимов В. В.

подпись, дата Ф. И .О.

Группа МКИС23

Программа профессиональной переподготовки: «Искус ственный интеллект и машинное обучение» (256ч.)

Руководитель

практической

подготовки от ДГТУ: Доцент Скляренко А.А.

должность Ф. И. О

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись руководителя от ДГТУ

Ростов-на-Дону

2023 г.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Подразделение, ответственное за реализацию ДПП ПП «Искусственный интеллект и машинное обучение» - Центр «Цифровая культура»

**ЗАДАНИЕ**

По практической подготовке при проведении практики в ООО «ИТР-ГРУПП»

в период с «17» апреля 2023 г. по «13» мая 2023 г.

Обучающийся Максимов Вадим Вячеславович

Группа МКИС23

Срок представления отчета в ДГТУ «13» мая 2023 г.

Содержание индивидуального задания:

Распознавание эмоций человека по фото на основе моделей глубокого обучения.

|  |  |  |
| --- | --- | --- |
| Руководитель практической подготовки от ДГТУ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Доцент А.А. Скляренко  должность. И.О.Ф. |
|  |  |  |
| Задание принял к исполнению | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | В.В. Максимов  имя, отчество, фамилия |
|  |  |  |

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИЧЕСКОЙ ПОДГОТОВКИ

|  |  |  |
| --- | --- | --- |
| Дата | Место работы/  практики | Выполняемые работы |
| 17.04.2023 | ООО «ИТР-ГРУПП» | Знакомство с предприятием, прохождение вводного инструктажа. |
| 17.04.2023 | ООО «ИТР-ГРУПП» | Ознакомление с территорией предприятия, прохождение первичного инструктажа по ТБ, ПБ |
| 18.04.2023 | ООО «ИТР-ГРУПП» | Получение индивидуального задания |
| 19.04.2023 – 22.04.2023 | ООО «ИТР-ГРУПП» | Аналитический обзор предметной области, постановка задачи |
| 24.04.2023 – 30.04.2023 | ООО «ИТР-ГРУПП» | Формирование датасета, выбор модели и определение ее архитектуры |
| 2.05.2023 – 8.05.2023 | ООО «ИТР-ГРУПП» | Программная реализация |
| 10.05.2023 | ООО «ИТР-ГРУПП» | Тестирование модели |
| 11.05.2023-12.05.2023 | ООО «ИТР-ГРУПП» | Подготовка и оформление отчета по практике |
| 13.05.2023 | ООО «ИТР-ГРУПП» | Сдача итогового отчета |

ОТЗЫВ - ХАРАКТЕРИСТИКА

Обучающийся Максимов Вадим Вячеславович

фамилия, имя, отчество

2 курса группы МКИС23

Наименование места практической подготовки

ООО «ИТР-ГРУПП»

наименование предприятия

Обучающийся по программе дополнительного профессионального образования профессиональной переподготовки «Искусственный интеллект и машинное обучение» выполнил задания, предусмотренные рабочей программы практической подготовки в полном объёме.

Дополнительно ознакомился/изучил

1. Методы машинного обучения
2. Методы глубокого обучения
3. Библиотеки для python NumPy, pandas, matlibplot, seaborn, OpenCV, pillow, TensorFlow

Заслуживает оценки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | Руководитель практической подготовки  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

**Содержание**

Введение 7

1 Аналитический обзор предметной области 8

1.1 Обзор задачи классификации изображений 8

1.2 Обзор методов машинного обучения для классификации изображений 8

1.3 Постановка задачи 8

2 Описание модели классификации изображений 9

2.1 Обоснование выбора модели 9

2.2 Архитектура модели 9

3 Программная реализация приложения 10

3.1 Обоснование выбора средств разработки 10

3.2 Описание программной реализации 10

4. Тестирование модели приложения 11

4.1 Описание процесса тестирования 11

Заключение 12

Перечень использованных информационных ресурсов 13

Приложение А Листинг программы 14

**Введение**

Задача распознавания изображений является очень важной, так как возможность автоматического распознавания компьютером изображений приносит множество новых возможностей в развитии науки и техники, таких, как разработка систем поиска лиц и других объектов на фотографиях, контроля качества производимой продукции без участия человека, автоматического управления транспортом и множество других.

В настоящее время для решения задачи классификации широко используются технологии искусственного интеллекта.

Поэтому целью данной работы является реализация web-приложения с использованием модели глубокого обучения для классификации изображений эмоций человека. Данную модель можно будет интегрировать в приложения, связанные с распознаванием эмоций человека на изображении.

**1 Аналитический обзор предметной области**

**1.1 Обзор задачи классификации изображений**

Предметная область, в которой необходимо создать модель обучения – это распознавание человеческих эмоций на фото. Задача состоит в том, что - что бы распознать какая эмоция изображена на фото и предоставить ответ в текстовом формате. Но для того, чтобы распознавание было гибкое для разных типов фото, должны быть выделены различные факторы.

Порой на одном и том же выражении лица может изображено несколько эмоций (к примеру, на фото рисунка 1.1 с нейтральным выражением лица можно классифицировать, как и грустное). Наша модель будет предсказывать только одну эмоцию.



Рисунок 1.1 - Пример неясного выражения лица

Также надо учитывать возраст человека, так как у одного и того человека в разные периоды времени может быть разная физиономия и черты лица. Черты лица ребёнка значительно отличаются от черт лица взрослого или пожилого человека (к примеру, на рисунке 1.2 изображено изменение физономии лица человека в течении времени).



Рисунок 1.2 - Пример разной физиономии лица в разные периоды жизни

Пол человека также имеет значение. Физиономия лиц мужчин и женщин заметно отличается (к примеру, у большинства женщин область подбородка более узкая, чем у мужчин).

Лицо может быть направлено не прямо на камеру, а куда-то в сторону. Этот фактор тоже надо учитывать при тренировке модели (к примеру, как показано на рисунке 1.3).



Рисунок 1.3 - Пример лица направленного в другую сторону

Лицо может быть прикрыто чем-то (к примеру, это могут быть волосы, как показано на рисунке 1.4, руки человека, элементы одежды и прочие предметы), из-за этого на фото будет изображена лишь неполная часть лица. Модель должна уметь обрабатывать фото и такого формата.



Рисунок 1.4 - Пример лица, прикрытого волосами

На фото может быть изображено два и больше лиц. Для нашей модели это может стать проблемой (к примеру, как показано на рисунке 1.5).



Рисунок 1.5 - Пример фото с несколькими лицами

Для решения этой проблемы можно создать новую модель нейросети для распознавания лиц людей и выводить названия распознанной эмоции прямо на фото рядом с лицом. Но из-за ограничений по времени выполнения практики нам придётся отказаться от этой идеи. Следовательно наша модель лучше будет работать с фото, где есть только одно лицо.

Также на изображении может изображено лицо человека не во всё изображение, а занимать лишь небольшую часть изображения, и быть где-то в стороне. (к примеру, как показано на рисунке 1.6).



Рисунок 1.6 - Пример фото, где лицо изображено не в центре

На фото может быть изображено не настоящий человек, а нарисованный или мультяшный персонаж. Если рисунок не реалистичный, это может вызвать некоторые проблемы (к примеру, если лицо будет круглое, как показано на рисунке 1.7, модели будет трудно характеризовать пол персонажа, так как явных признаков для это не будет). Поэтому мы снабдим датасет для обучения некоторыми нарисованными картинками.



Рисунок 1.7 - Пример нарисованного изображения с нестандартной физиономией лица

Цветовая гамма повлияет на распознавание, поэтому на все картинки будет применён один тот же цветовой фильтр. Размер и разрешение картинки тоже может сыграть свою роль(к примеру, как показано на рисунке 1.8). В тренировочном датасете все картинки будут одного размера, а иные картинки будут масштабироваться под нужный размер в самой программе до предсказывания эмоции. Ссылка на датасет - <https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer>.



Рисунок 1.8 - Пример разных цветовых фильтров на фото

**1.2 Обзор методов машинного обучения для классификации изображений**

В настоящее время выделяется две группы методов машинного обучения для классификации изображений. Первая группа основывается на методах классического машинного обучени. К первого группе относят:

Регрессия помогает нам находить корреляцию между переменными и позволяет прогнозировать непрерывные выходные переменные на основе одной или нескольких переменных-предикторов.

Кластеризация организует сопоставимые точки данных в кластеры или подгруппы на основе их внутреннего сходства. Это задача для обучения без учителя.

Классификация в машинном обучении — это процесс группирования объектов по категориям на основе предварительно классифицированного тренировочного набора данных.

Вторая группа методов относится к методам глубокого обучения. К ней относят:

Рекуррентная нейронная сеть (RNN) может обрабатывать последовательности данных, таких как текст или аудио. Но она не выполняет задачи распознавания эмоций на фото, поэтому этот вид модели не будет применятся.

Трансформационная нейронная сеть (Transformer) может обрабатывать большие объёмы данных и может использоваться для анализа текстовых данных. Ситуация, как и с RNN – нам необходима обработка графических изображений, а не текста.

Сверточная нейронная сеть (CNN) используется для обработки изображений и может быть адаптирована для работы с аудио или видео данными. Сверточная нейронная сеть может распознавать эмоции по мимике лица, движении глаз и другими признаками, которые могут быть получены из данных изображения. Эта модель нейронной сети удовлетворяет требованиям.

**1.3 Постановка задачи**

Проведенный обзор предметной области показал целесообразность и эффективность применения методов машинного обучения, а именно глубокого обучения, для классификации изображений. Поэтому целью данной работы является реализация модели глубокого обучения для классификации изображений знаков.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* сформировать обучаемый датасет, состоящий из множества изображений различных дорожных знаков.
* выбрать модель глубокого обучения для классификации изображений;
* обучить модель;
* провести тестирование обученной модели;
* применить модель для практичексих примеров.
* разработать backend по обработке изображения для её распознавания моделью
* разработать frontend для загрузки изображения и отображения результата

**2 Описание модели классификации изображений**

**2.1 Обоснование выбора модели**

Регрессия не классифицирует данные и необходима для предсказания результата на основе предыдущих данных. Значит этот метод не подходит.

В нашем случае для ускорения процесса обучения будет использовано обучение с учителем. Поэтому метод кластеризации не будет использоваться в модели.

Так как у датасет разделён на архивы из 7 эмоций, метод классификации на основе сверточных нейроных сетей идеально подходит для обучения, и он будет использован в тренировке модели. Сверточная нейронная сеть позволит выделить признаки на изображениях и далее провести классификацию по ним.

**2.2 Архитектура модели**

В качестве входных данных модель принимает картинку, преобразованную в матрицу. В ней хранятся данные о цветах в числовых значениях.

Архитектура проекта имеет следущие элементы:

Dense – стандартный слой, который хранит в себе нейроны.

Conv2D – слой, используемый свёрточной сетью. Необходим для создания ядра свёртки.

Dropout – немного изменяет входные параметры при тренировке модели.

Flatten – слой, который преобразует матрицу в вектор.

Batch Normalization – нормализует данные при тренировке, уменьшая разницу между ними.

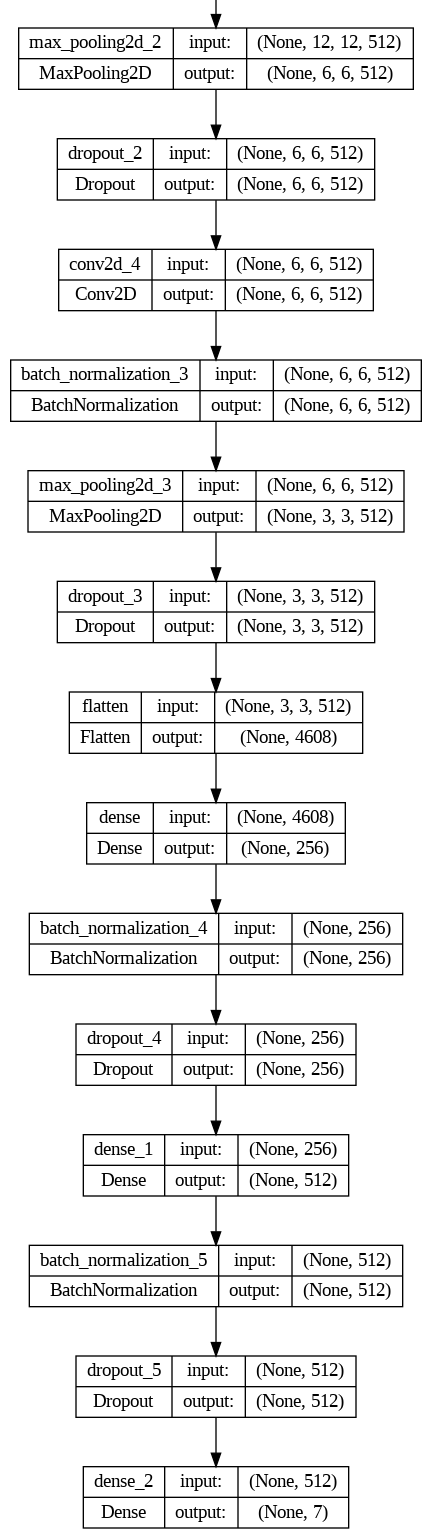
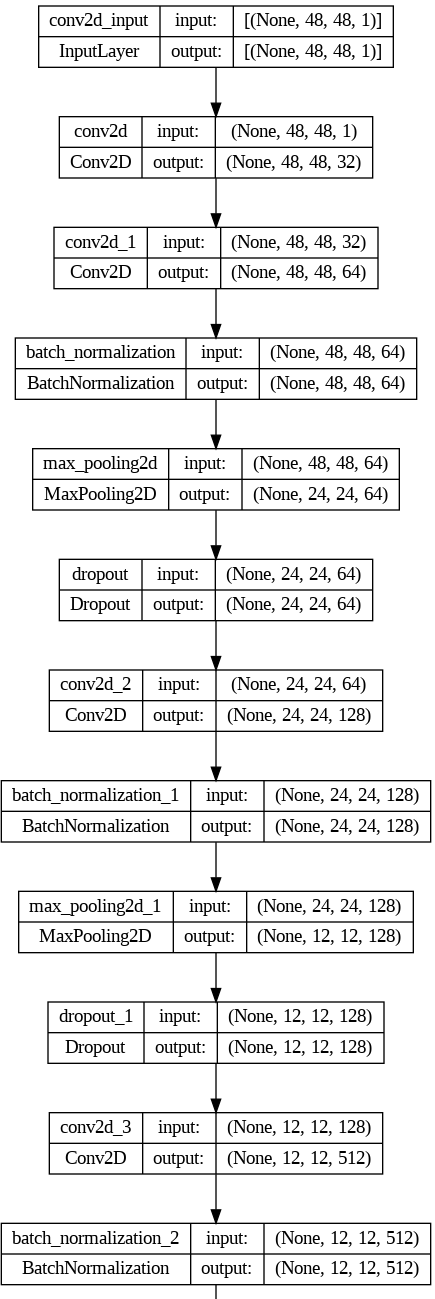


Рисунок 2.1 – Архитектура сверточной нейросети

**2.3 Алгоритм применения модели для классификации изображений**

Для применения обученной модели необходимо выполнить следующую последовательность действий:

1) Пользователь загружает своё фото на web-страницу

2) Backend обрабатывает загруженное пользователем изображение: задает ему необходимый масштаб, размер и цветовой фильтр.

3) Обработанное изображение проходит через натренированную модель.

4) Интерпретация выходных данных модели

5) Программа выдаёт загруженное изображение, распознаную эмоцию и процент истинности на web-странницу.

**3 Программная реализация приложения**

**3.1 Обоснование выбора средств разработки**

Программа будет реализована на языке программирования Python. Тренировка модели будет реализована с помощью библиотеки для Искусственного интеллекта TensorFlow.

Приложение, которое будет использовать натренированную модель будет реализовано на языке программирования Python с использованием библиотеки для создания веб приложения Django.

Визуальный стиль приложения (Frontend) будет реализован на языке разметки HTML5 и CSS.

TensoFlow был выбран по следующим критериям:

1. Многофункциональный
2. Имеет свой фреймворк для развертывания в produсtion.
3. Хорошая поддержка под мобильные устройства.
4. Подробная документация.

Django был выбран по следующим критериям:

1. Требуется быстро работать, быстро развертывать и вносить изменения в проект по ходу работы.
2. В любой момент в приложении может потребоваться масштабирование: как наращивание, так и сокращение.
3. Необходимо интегрировать новейшие технологии, например, машинное обучение.

Для обработки будет использоваться библиотека работы с изображениями для языка программирования Python Open-CV.

**3.2 Описание программной реализации**

Проект использует структуру проекта Django. С помощью единственного приложения можно загрузить изображения, программа его обрабатывает и выводит web-страницу. Проект использует локальную сеть. Для запуска программы необходимо:

1. Перейти в папку проекта через консоль.
2. Подключить виртуальную среду.
3. Переёти папку проекта Django

cd testing

1. Запустить локальный сервер

python manage.py runserver

Приложение имеет слудущю структуру:

* models.py – создаёт структуру проекта Django, в которой сохраняется название изображения и само изображения. Все данные, которые были сохранены во время работы локального сервера, удаляются после его отключения.
* form.py – создаёт форму для сохранения данных в модель.
* views.py – в ней происходит все основные действия:

Функция prepare принимает в кацестве единтвенного параметра путь к сохраненому файлу. Она преобразует изображение в матрицу и сохраняет его в переменную.

Функция image\_upload\_view позволяет загрузить и сохранить через форму изображение. Затем загруженное изображение обрабатывается с помощью функции prepare и проходит через заранее обученную модель. Затем результат сохраняется в словарь и загружается на web-страницу. В словаре хранится: название изображения, само изображение и результат распознаной эмоции.

**4. Тестирование приложения**

**4.1 Описание процесса тестирования**

Для начала загрузим изображение из датасета тренировки (рис. 4.1). Результат был выведен верным.

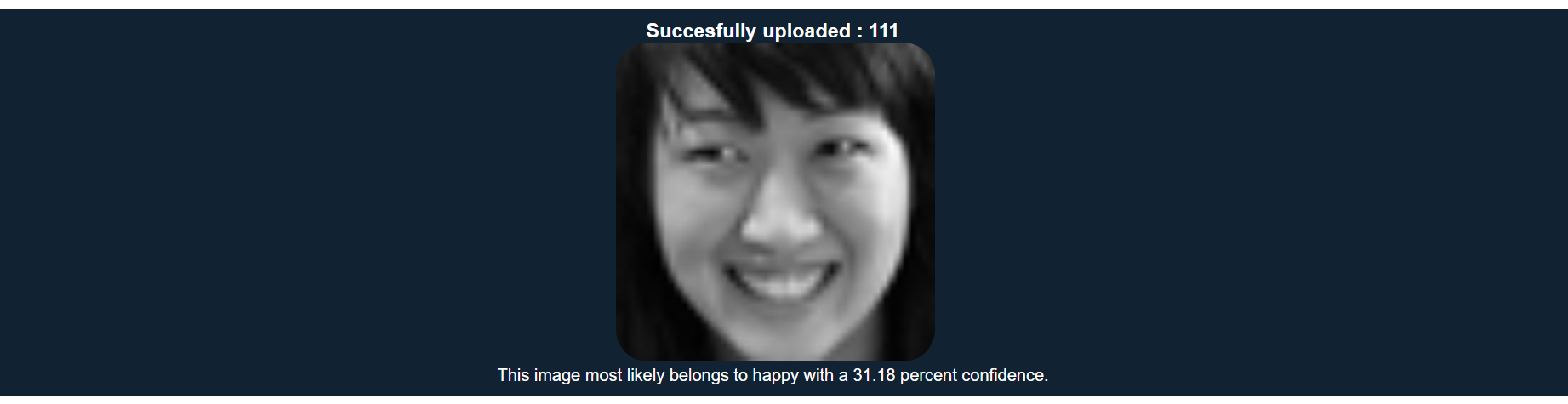


Рисунок 4.1 – Изображение из обучаемого датасета

Следующим будет тестироватся изображение не из датасета и другого цветового фильтра (рис. 4.2). Результат частично верный, картинку можно классифицровать как neutral (нейтральный).

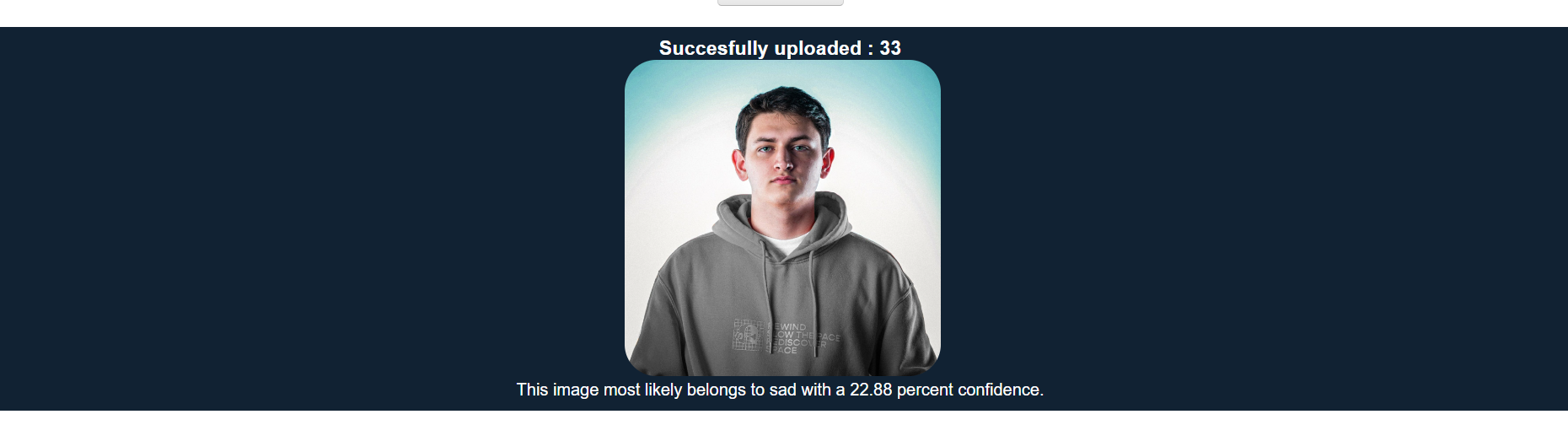


Рисунок 4.2 – Стороннее изображение отличного цветового фильтра

**Заключение**

В ходе выполнения данной работы была выбрана и реализована модель для классификации изображений эмоций человека, которая позволяет автоматизировать процесс распознавания эмоции на фото и может быть использована в различных практико-ориентированных приложениях.

Получены и закреплены навыки работы с моделями: выбор, реализация, обучение, тестирование и возможность применения для решения практических задач.

Кроме этого, отработаны навыки формирования датасета и оценки возможности его использования для обучения моделей.

**Перечень использованных информационных ресурсов**

1. Ссылка на тренировочный датасет [Электронный ресурс] URL: <https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer> (дата обращения 20.05.2023)

2. Сcылка к документации Django [Электронный ресурс] URL: <https://docs.djangoproject.com/en/4.2/> (дата обращения 22.05.2023)

3. Ссылка на тренирвочную программу для модели нейросети [Электронный ресурс] URL: <https://drive.google.com/file/d/1H7RrnSWBd5W0aZKeA3InBjl_UA9HK4Ca/view?usp=sharing> (дата обращения 25.05.2023)

**Приложение А Листинг программы**

1. views.py

from django.shortcuts import render

from .forms import ImageForm

import numpy as np

import tensorflow as tf

import keras.utils as image

import cv2

def prepare(filepath):

IMG\_SIZE = 48

img\_array = cv2.imread(filepath)

new\_array = cv2.resize(img\_array, (IMG\_SIZE, IMG\_SIZE))

new\_array = new\_array.reshape(-1, IMG\_SIZE, IMG\_SIZE, 3)

new\_array = new\_array.astype("float") / 255.0

return new\_array

def image\_upload\_view(request):

"""Process images uploaded by users"""

if request.method == 'POST':

form = ImageForm(request.POST, request.FILES)

if form.is\_valid():

form.save()

# Get the current instance object to display in the template

img\_obj = form.instance

asd = str(request.FILES)[51:-16]

model = tf.keras.models.load\_model('./models/emotion\_detection.h5')

CATEGORIES = ['angry', 'disgusted', 'fearful', 'happy', 'neutral', 'sad', 'surprised']

prediction = model.predict([prepare("./media/images/" + asd)])

#prediction # will be a list in a list.

score = tf.nn.softmax(prediction[0])

answer = "This image most likely belongs to {} with a {:.2f} percent confidence.".format(CATEGORIES[np.argmax(score)], 100 \* np.max(score))

#return render(request, 'index.html', {'form': form, 'img\_obj': img\_obj})

return render(request, 'index.html', {'form': form, 'img\_obj': img\_obj, 'answer': answer})

else:

form = ImageForm()

return render(request, 'index.html', {'form': form})

2. models.py

from django.db import models

class Image(models.Model):

title = models.CharField(max\_length=200)

image = models.ImageField(upload\_to='images')

def \_\_str\_\_(self):

return self.title

3. forms.py

from django import forms

from .models import Image

class ImageForm(forms.ModelForm):

"""Form for the image model"""

class Meta:

model = Image

fields = ('title', 'image')