СВЕТОЧ
Образовательный центр

ПРОГРАММА
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

“Анализ данных с использованием нейросетей в DATA Science”

Преподаватель курса
Ахмедов Марлен Игоревич

Выполнил
Авдеев Михаил Иванович

№ потока
DS(144)_23-2.2

Москва 2023 г.

СОДЕЙСТВИЕ ЗАНЯТОСТИ | Федеральный проект

Проект по теме
“Обработка фотографий людей с помощью нейронной сети и детекция среди них индивидуумов с отсутствием волос на волосистой части головы”

Содержание презентации:

- Введение
- Сбор и подготовка данных
- Выбор архитектуры модели
- Обучение модели
- Оценка модели
- Визуализация результатов
- Документация и отчет
- Презентация проекта, его результаты
- Дальнейшие улучшения

Введение

- Краткое описание проекта и его цели:

  Проект направлен на разработку системы, способной обрабатывать фотографии волосистой части головы человека и определять проблемные участки.

- Зачем нужна обработка фотографий людей и детекции проблем на волосистой части головы:

  Обработка таких фотографий позволит, при должном развитии проекта, автоматически выявлять и классифицировать различные виды алопеции, изменения состояния волос и кожи головы для создания приложения по диагностике и подбору рекомендаций по лечению волос и уходу за ними.

Сбор и подготовка данных

- Данные представляют собой фотографии людей разделённых по принципу есть волосы на голове/нет волос.
- Данные были собраны из публичных источников.
- Подготовка данных, масштабирование и обработка изображений:

  Разметка данных производилась с присваиванием имён файлам по принципу bald/not_bald с добавлением порядковых номеров по 100 шт.

  Используя библиотеку TensorFlow и Keras, изображения были масштабированы, чтобы значения пикселей находились в диапазоне от 0 до 1. Это помогало модели обрабатывать изображения более эффективно и улучшать результаты классификации.

Выбор архитектуры модели

В данном проекте была выбрана архитектура сверточной нейронной сети (Convolutional Neural Network, CNN), так как она демонстрирует хорошую производительность в задачах обработки изображений.

Были рассмотрены различные слои:

- Conv2D (сверточные слои)
- MaxPooling2D (слои пулинга)
- Flatten (слой преобразования)
- Dense (полносвязные слои)

Использовалась активационная функция relu для повышения нелинейности

Слои и их параметры определены в Sequential модели.

```
model = keras.Sequential([
    keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Conv2D(64, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Conv2D(128, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Conv2D(128, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Flatten(),
    keras.layers.Dense(512, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')])
```

Выбранная архитектура сверточной нейронной сети (CNN) позволяет модели эффективно обрабатывать изображения и извлекать важные признаки, необходимые для классификации отсутствия волос на волосистой части головы. CNN хорошо подходит для задач распознавания образов и позволяет сети автоматически изучать различные характеристики изображений.

Обучение модели

Модель обучается с использованием функции потерь бинарной классификации (binary_crossentropy) и оптимизатора Adam (adam).

Функция потерь позволяет оценить, насколько хорошо модель классифицирует изображения, а оптимизатор позволяет обновлять веса модели для минимизации потерь.

```
# Компилируем модель
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Обучаем модель
history = model.fit(
    train_generator,
    steps_per_epoch=len(train_generator),
    epochs=10,
    validation_data=test_generator,
    validation_steps=len(test_generator))
```

Для эффективного использования данных и предотвращения переобучения модели, были использованы генераторы изображений для загрузки обучающих и тестовых данных по частям. Генераторы позволили модели обрабатывать данные пакетами и автомоматически масштабировать и аугментировать изображения для повышения разнообразия данных и улучшения обобщающей способности модели.

Параметры обучения модели - количество эпох обучения, шаги на эпоху, размер пакета и другие параметры были определены для достижения оптимальных результатов обучения модели. Эти параметры могут быть изменены и оптимизированы для конкретных потребностей проекта.

Оценка модели

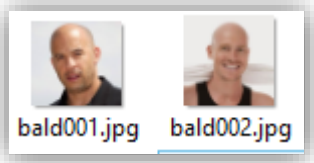
Модель была оценена на обучающих и тестовых данных с помощью метрики точности (accuracy) и функции потерь (loss).
Точность показывает, насколько хорошо модель классифицирует изображения, а потери отражают, насколько хорошо модель соответствует эталонным меткам.
Были получены результаты точности и потерь модели на обучающих и тестовых данных, что позволило оценить способность модели обобщать и классифицировать изображения соответственно.

```
Found 40 images belonging to 2 classes.
Epoch 1/10
8/8 [==============================] - 20s 2s/step - loss: 0.7189 - accuracy: 0.5562 - val_loss: 0.6889 - val_accuracy: 0.5000
Epoch 2/10
8/8 [==============================] - 18s 2s/step - loss: 0.6403 - accuracy: 0.7250 - val_loss: 0.5287 - val_accuracy: 0.7500
Epoch 3/10
8/8 [==============================] - 18s 2s/step - loss: 0.6023 - accuracy: 0.7250 - val_loss: 0.4279 - val_accuracy: 0.8500
```

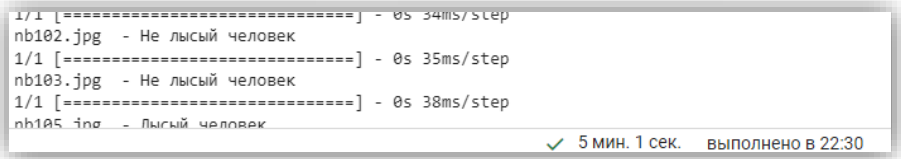
Эти результаты будут использованы для дальнейшего улучшения модели и оптимизации ее параметров.
Используя эту информацию, можно определить, насколько хорошо модель справляется с задачей детекции отсутствия волос на волосистой части головы.

Визуализация результатов

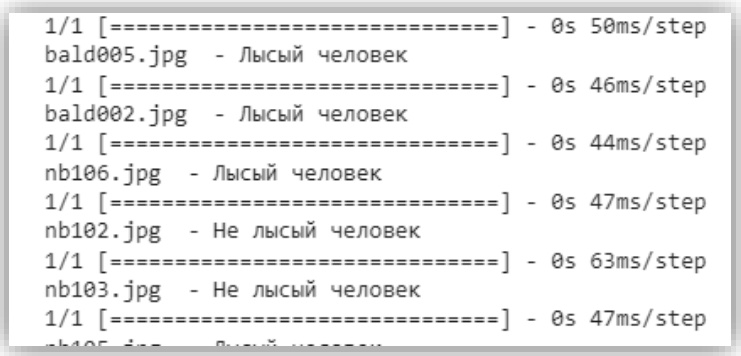
Примеры фотографий составляющих базу данных.



Время обучения модели на предложенной выборке и проведение детекции контрольных фотографий.



Демонстрация работы модели по выборочным контрольным фотографиям – время на детекцию и сам результат детекции.



Документация и отчет

Проект по теме “Обработка фотографий людей с помощью нейронной сети и детекция среди них людей с отсутствием волос на волосистой части головы”

Описание:
Проект направлен на разработку системы, способной обрабатывать фотографии волосистой части головы человека и определять проблемные участки, используя и обучая свёрточную нейронную сеть и библиотеки TensorFlow и Keras.

Этапы разработки проекта:

1. Сбор и подготовка данных – собраны и обработы фотографии людей из публичных источников с дифференциацией по признаку есть волосы на голове/нет волос на голове по 100 шт.
2. Выбор архитектуры модели - была выбрана архитектура сверточной нейронной сети с использованием слоёв свёрточных, пулинга, полносвязных и слоёв преобразования.
3. Обучение модели - модель обучается с использованием функции потерь бинарной классификации (binary_crossentropy) и оптимизатора Adam (adam).
4. Оценка производительности модели на выбранных данных - модель была оценена на обучающих и тестовых данных с помощью метрики точности (accuracy) и функции потерь (loss).

Документация и отчет

Проект по теме “Обработка фотографий людей с помощью нейронной сети и детекция среди них людей с отсутствием волос на волосистой части головы”

Данный проект является основой для разработки приложения для оценки состояния здоровья волосистой части головы и позволяет получить вектор для обучения более продвинутой модели, способной определять различные проблемы волосистой части головы человека.

При должной проработке проекта приложение сможет, совместно с анкетированием пользователя, определять очаговую (гнёздную) алопецию, диффузную алопецию, андрогенную алопецию, состояние волосяных фоликул и другие поталогии волос и кожи волосистой части головы.

Презентация проекта, его результаты

База данных:
Количество фотографий людей с волосами на волосистой части головы – 100 шт.
Количество фотографий людей без волос на волосистой части головы – 100 шт.
Соотношений выборки обучающей и тестовой в % - 80/20
Архитектура нейросети:
Структура данной модели состоит из нескольких сверточных слоев, слоев пулинга, плоского слоя и двух полносвязанных слоев, применяемых для классификации.
В первом сверточном слое `Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3))` используется 32 фильтра размером 3x3.
Во втором сверточном слое `Conv2D(64, (3, 3), activation='relu')` используется 64 фильтра размером 3x3.
Таким образом, количество цифр указывает на сложность модели и ее способность выделять более высокоуровневые особенности из изображений.
После сверточных слоев следуют слои пулинга `MaxPooling2D`, которые уменьшают размерность данных, удаляя избыточную информацию и улучшая вычислительную эффективность. В данном коде используется пулинг размером (2, 2), что означает уменьшение размера каждой размерности в два раза.
После пулинга следуют еще два сверточных слоя и два слоя пулинга с теми же параметрами. Затем идет плоский слой `Flatten`, который преобразует многомерные данные в одномерный массив.
Далее идут два полносвязанных слоя (`Dense`), которые преобразуют данные, применяя линейные операции и функции активации.
В данном случае, первый полносвязанный слой имеет 512 нейронов и использует функцию активации 'relu', а второй полносвязанный слой состоит из одного нейрона с сигмоидной функцией активации.

Презентация проекта, его результаты

Результаты:

Обучение на 10 эпохах с размером пакета 20

Потеря min/max – 0,0710/0,8471

Точность min/max – 0,5063/0,9625

Время обучения на предложенной базе данных – 5 мин. 1 сек.

Вероятность положительной детекции – в среднем 70%

```
acy: 0.8750
1/1 [==============================] - 0s 156ms/step
bald004.jpg  - Лысый человек
1/1 [==============================] - 0s 36ms/step
bald100.jpg  - Не лысый человек
1/1 [==============================] - 0s 34ms/step
bald003.jpg  - Лысый человек
1/1 [==============================] - 0s 41ms/step
bald001.jpg  - Лысый человек
1/1 [==============================] - 0s 35ms/step
bald099.jpg  - Не лысый человек
1/1 [==============================] - 0s 36ms/step
bald005.jpg  - Лысый человек
1/1 [==============================] - 0s 38ms/step
bald002.jpg  - Лысый человек
1/1 [==============================] - 0s 34ms/step
nb106.jpg  - Не лысый человек
1/1 [==============================] - 0s 34ms/step
nb102.jpg  - Не лысый человек
1/1 [==============================] - 0s 35ms/step
nb103.jpg  - Не лысый человек
1/1 [==============================] - 0s 38ms/step
nb105.jpg  - Лысый человек
1/1 [==============================] - 0s 35ms/step
nb104.jpg  - Не лысый человек
1/1 [==============================] - 0s 35ms/step
nb108.jpg  - Не лысый человек
1/1 [==============================] - 0s 37ms/step
nb101.jpg  - Не лысый человек
1/1 [==============================] - 0s 35ms/step
nb107.jpg  - Не лысый человек
1/1 [==============================] - 0s 40ms/step
bald101.jpg  - Не лысый человек
```

Дальнейшие улучшения

Увеличение и проработка базы данных:
Необходимо конечно же увеличить количество фотографий в базе данных и уделить внимание их проработке и приведению в должное состояние (минимум 1000 шт. каждого варианта bald/not_bald).

Анализ и коррекция Гиперпараметров:
Исследование влияния изменений параметров, таких как количество эпох, размер пакета, и других гиперпараметров на производительность модели. (Например использование Keras тюнер).

Анализ и внедрение дополнительных слоёв и архитектурных изменений:
Рассмотрение добавления дополнительных сверточных и полносвязных слоев, а также изменения архитектуры для повышения точности.

Использование предобученных моделей:
Можно проверить применение предобученных моделей, такие как VGG16, ResNet или Inception, и дообучить их на своих данных. Предобученные модели уже обучены на больших наборах данных и способны извлекать высокоуровневые признаки из изображений.

Дальнейшие улучшения

Пример использования предобученной модели Inception

Ещё не получено улучшение точности, но для начала необходимо увеличить и улучшить базу данных.

```
# Создаем модель нейронной сети
model = keras.Sequential()
model.add(keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)))
model.add(keras.layers.MaxPooling2D((2, 2)))
model.add(keras.layers.Conv2D(64, (3, 3), activation='relu'))
model.add(keras.layers.MaxPooling2D((2, 2)))
model.add(keras.layers.Conv2D(128, (3, 3), activation='relu'))
model.add(keras.layers.MaxPooling2D((2, 2)))
model.add(keras.layers.Conv2D(128, (3, 3), activation='relu'))
model.add(keras.layers.MaxPooling2D((2, 2)))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(512, activation='relu'))
model.add(keras.layers.Dense(1, activation='sigmoid'))

# Добавляем модель Inception в нашу модель
inception_model = keras.applications.InceptionV3(weights='imagenet', include_top=False, i
inception_model.trainable = False

# Компилируем модель
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

```
Epoch 1/10
8/8 [==============================] - 73s 9s/step - loss: 0.6863 - accuracy: 0.5688 - val_loss: 0.6891 - val_accur
acy: 0.5000
Epoch 2/10
8/8 [==============================] - 17s 2s/step - loss: 0.7182 - accuracy: 0.5375 - val_loss: 0.5606 - val_accur
acy: 0.7750
Epoch 3/10
8/8 [==============================] - 17s 2s/step - loss: 0.5971 - accuracy: 0.7063 - val_loss: 0.5690 - val_accur
acy: 0.8000
Epoch 4/10
8/8 [==============================] - 18s 2s/step - loss: 0.4958 - accuracy: 0.7250 - val_loss: 0.4452 - val_accur
acy: 0.8250
Epoch 5/10
8/8 [==============================] - 18s 2s/step - loss: 0.3896 - accuracy: 0.8313 - val_loss: 0.6699 - val_accur
acy: 0.7000
Epoch 6/10
8/8 [==============================] - 18s 2s/step - loss: 0.2685 - accuracy: 0.9125 - val_loss: 0.2981 - val_accur
acy: 0.8750
Epoch 7/10
8/8 [==============================] - 17s 2s/step - loss: 0.1810 - accuracy: 0.9438 - val_loss: 0.4847 - val_accur
acy: 0.8250
Epoch 8/10
8/8 [==============================] - 18s 2s/step - loss: 0.1169 - accuracy: 0.9500 - val_loss: 0.4988 - val_accur
acy: 0.8250
Epoch 9/10
8/8 [==============================] - 18s 2s/step - loss: 0.1475 - accuracy: 0.9375 - val_loss: 0.3424 - val_accur
acy: 0.9000
Epoch 10/10
8/8 [==============================] - 17s 2s/step - loss: 0.1050 - accuracy: 0.9563 - val_loss: 0.4742 - val_accur
acy: 0.8750
```

```
1/1 [==============================] - 0s 126ms/step
bald004.jpg  - Лысый человек
1/1 [==============================] - 0s 32ms/step
bald100.jpg  - Не лысый человек
1/1 [==============================] - 0s 33ms/step
bald003.jpg  - Лысый человек
1/1 [==============================] - 0s 51ms/step
bald001.jpg  - Лысый человек
1/1 [==============================] - 0s 34ms/step
bald099.jpg  - Не лысый человек
1/1 [==============================] - 0s 32ms/step
bald005.jpg  - Лысый человек
1/1 [==============================] - 0s 32ms/step
bald002.jpg  - Лысый человек
1/1 [==============================] - 0s 49ms/step
nb106.jpg  - Не лысый человек
1/1 [==============================] - 0s 61ms/step
nb102.jpg  - Лысый человек
1/1 [==============================] - 0s 55ms/step
nb103.jpg  - Лысый человек
1/1 [==============================] - 0s 53ms/step
nb105.jpg  - Лысый человек
1/1 [==============================] - 0s 51ms/step
nb104.jpg  - Не лысый человек
1/1 [==============================] - 0s 47ms/step
nb108.jpg  - Не лысый человек
1/1 [==============================] - 0s 49ms/step
nb101.jpg  - Не лысый человек
1/1 [==============================] - 0s 48ms/step
nb107.jpg  - Не лысый человек
1/1 [==============================] - 0s 55ms/step
bald101.jpg  - Не лысый человек
```

СВЕТОЧ
Образовательный центр

Благодарю за внимание!

info@eduom.ru