

СЕКЦИЯ VIII ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И МОДЕЛИРОВАНИЕ

АВТОМАТИЗИРОВАННАЯ СИСТЕМА КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ ТКАНЕЙ ПАРАЩИТОВИДНОЙ ЖЕЛЕЗЫ НА ОСНОВЕ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Д. А. Давидовский

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Н. Н. Масалитина

Информационные технологии в настоящее время используются повсеместно. Большое количество стран уже давно активно используют новые технологии в сфере здравоохранения. Проведение консультаций пациентов и персонала дистанционно, обмен информацией о пациентах между различными учреждениями, фиксирование физиологических параметров, контроль за проведением операций в реальном времени – внедрение ИТ позволяет осуществлять все эти возможности. Это выводит здравоохранение на новый уровень развития, положительно сказываясь на всех аспектах его деятельности.

Главной задачей является классификация изображений с пораженными клетками парашитовидной железы и со здоровыми. С такой задачей хорошо справляется искусственная нейронная сеть. Но для того чтобы она решила эту задачу, ей необходимо на вход передать некоторые дескрипторы, по которым она будет принимать решение и классифицировать изображение. Контуры клеток парашитовидной железы являются характеристическим признаком для изображений.

Один из наиболее известных способов кодирования контура – это цепной код Фримена. Цепные коды применяются для представления границы в виде последовательности отрезков прямых линий определенной длины и направления. В основе этого представления лежит четырех- или восьмисвязная решетка. Длина каждого отрезка определяется разрешением решетки, а направления задаются выбранным кодом (рис. 1).

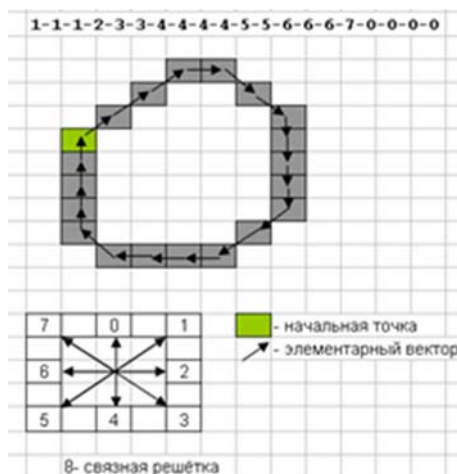


Рис. 1. Алгоритм «Цепной код Фримена»

На контуре фиксируется точка, которая называется начальной точкой. Затем контур обходится, и каждый вектор смещения записывается комплексным числом $a + ib$, где a – смещение точки по оси X , а b – смещение по оси Y . Смещение берется относительно предыдущей точки. При таком подходе контур определяется как совокупность элементарных векторов, представленных своими двумерными координатами.

Перед применением алгоритма контурного анализа изображения необходимо провести предварительную обработку: придать четкость границам, подавить шумы на изображении, преобразовать его в бинарный вид.

На рис. 2 показано изображение в оригинальном виде и после предварительной обработки.

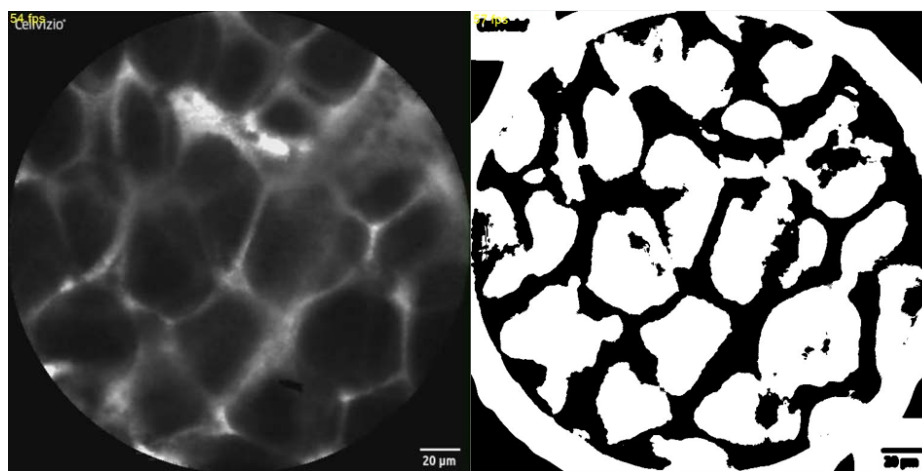


Рис. 2. Обработанное изображение

После выполнения алгоритма поиска контуров с помощью цепного ключа Фримена получается следующий результат (рис. 3).



Рис. 3. Результат выделения контуров

Полученный результат составляет основу для формирования выборки для обучения искусственной нейронной сети, которая будет классифицировать изображения с пораженными клетками и здоровыми.

Для решения поставленной задачи была взята предобученная сеть *Inception v3*. Архитектура сети *Inception v3* (рис. 4) способна решать похожие задачи классификации. *Inception v3* отработала с лучшими результатами точности классификации на соревновании *ImageNet 2015* и хорошо подходит для распознавания объектов. Отличительной особенностью данной архитектуры являются изменения размера ядра свертки с 5×5 на 3×3 и перестроение комбинации сверточных слоев. Для настройки работы сети под поставленную задачу ее необходимо дополнить слоем, который содержит 4 нейрона, соответствующих 4-м классам классификации.

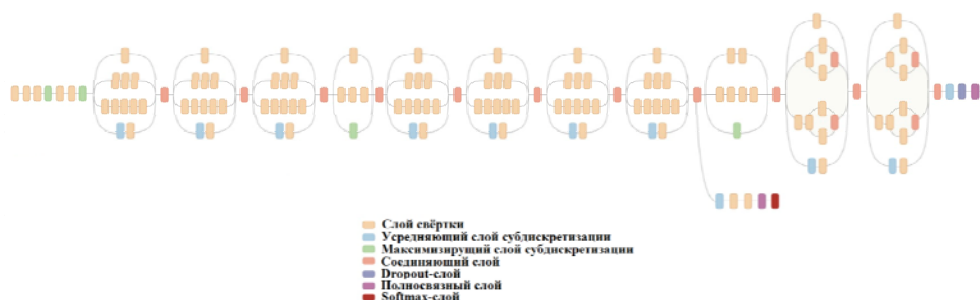


Рис. 4. Архитектура сети Inception v3

После обучения сети методом оптимизации стохастического градиентного спуска за 10 эпох на тестовой выборке она показала точность 59,52 %. Процесс обучения сети и результат показаны на рис. 5.

```
Epoch 1/10
57/57 [=====] - 281s 5s/step - loss: nan - acc: 0.5731 - val_loss: nan - val_acc: 0.5952
Epoch 2/10
57/57 [=====] - 258s 5s/step - loss: nan - acc: 0.5964 - val_loss: nan - val_acc: 0.5443
Epoch 3/10
57/57 [=====] - 257s 5s/step - loss: nan - acc: 0.5690 - val_loss: nan - val_acc: 0.6076
Epoch 4/10
57/57 [=====] - 265s 5s/step - loss: nan - acc: 0.5789 - val_loss: nan - val_acc: 0.5696
Epoch 5/10
57/57 [=====] - 285s 5s/step - loss: nan - acc: 0.5773 - val_loss: nan - val_acc: 0.5823
Epoch 6/10
57/57 [=====] - 267s 5s/step - loss: nan - acc: 0.5831 - val_loss: nan - val_acc: 0.6203
Epoch 7/10
57/57 [=====] - 250s 4s/step - loss: nan - acc: 0.5848 - val_loss: nan - val_acc: 0.5823
Epoch 8/10
57/57 [=====] - 250s 4s/step - loss: nan - acc: 0.5739 - val_loss: nan - val_acc: 0.6076
Epoch 9/10
57/57 [=====] - 254s 4s/step - loss: nan - acc: 0.5840 - val_loss: nan - val_acc: 0.5443
Epoch 10/10
57/57 [=====] - 248s 4s/step - loss: nan - acc: 0.6006 - val_loss: nan - val_acc: 0.5443

<tensorflow.python.keras._impl.keras.callbacks.History at 0xa7f457e160>

scores = modelSGD.evaluate_generator(test_generator, nb_test_samples // batch_size)
print("Аккуратность на тестовых данных: %.2f%%" % (scores[1]*100))

Аккуратность на тестовых данных: 59.52%
```

Рис. 5. Результаты обучения сети Inception v3

Обучаем эту же сеть методом оптимизации на основе градиента стохастических целевых функций первого порядка, основанным на адаптивных оценках моментов младшего порядка. Сеть показала лучшие результаты классификации за меньшее количество эпох (рис. 6).

```
Epoch 1/3
57/57 [=====] - 287s 5s/step - loss: 1.1888 - acc: 0.5480 - val_loss: 1.2410 - val_acc: 0.5833
Epoch 2/3
57/57 [=====] - 264s 5s/step - loss: 0.8613 - acc: 0.6992 - val_loss: 1.3709 - val_acc: 0.5570
Epoch 3/3
57/57 [=====] - 258s 5s/step - loss: 0.8835 - acc: 0.6716 - val_loss: 1.4202 - val_acc: 0.5696

<tensorflow.python.keras._impl.keras.callbacks.History at 0x993acee6d8>

scores = model.evaluate_generator(test_generator, nb_test_samples // batch_size)
print("Аккуратность на тестовых данных: %.2f%%" % (scores[1]*100))

Аккуратность на тестовых данных: 60.71%
```

Рис. 6. Результаты обучения сети Inception v3 за 3 эпохи

Обученные сети показали довольно хорошие показатели на небольшой тренировочной выборке.

Литература

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс // Техносфера. – М., 2005.