

САНКТ-ПЕТЕРБУРГСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра Системного Программирования  
Группа 20.Б11-мм

Азарников Иван Евгеньевич

Разработка мобильного приложения  
для определения рака кожи

Отчет по учебной практике  
в форме "Решение"

*Научный руководитель:*  
доцент кафедры СП, к.т.н. ЛИТВИНОВ Ю.В.

Санкт-Петербург  
2022

# Содержание

Введение	3
<b>1 Постановка задачи</b>	<b>4</b>
<b>2 Обзор</b>	<b>5</b>
2.1 MIRF . . . . .	5
2.1.1 Архитектура . . . . .	5
2.1.2 MIRFskinCancer . . . . .	6
2.2 Существующие решения . . . . .	6
2.3 Используемые инструменты . . . . .	7
<b>3 Ход работы</b>	<b>8</b>
3.1 Разработка графического интерфейса . . . . .	8
3.2 Разработка программной части приложения . . . . .	8
3.3 Тестирование приложения на наборе изображений . . . . .	9
3.4 Реализация нейронной сети . . . . .	10
<b>4 Заключение</b>	<b>12</b>

## Введение

За родинками необходимо следить. Происходящие с ними изменения могут быть первыми признаками рака кожи. Это одно из наиболее распространенных онкологических заболеваний по данным Всемирной организации здравоохранения [2]. Наиболее часто встречающимся типом этого вида рака является меланома.

По данным ВОЗ ежегодно регистрируется около 133000 новых случаев меланомы. Меланома наиболее агрессивна и чаще дает метастазы<sup>1</sup>. Продолжительность жизни при меланоме зависит прежде всего от стадии опухолевого прогресса. Выявление меланомы на начальной стадии по прогнозу наиболее перспективно для полного лечения. Поэтому важно обнаружить заболевание на ранней стадии, так как с ним проще справиться. Меланому легко спутать с родинкой, поэтому при появлении таковой лучше провериться у специалиста. На Рис. 1<sup>2</sup> представлен пример злокачественной и доброкачественной родинок.

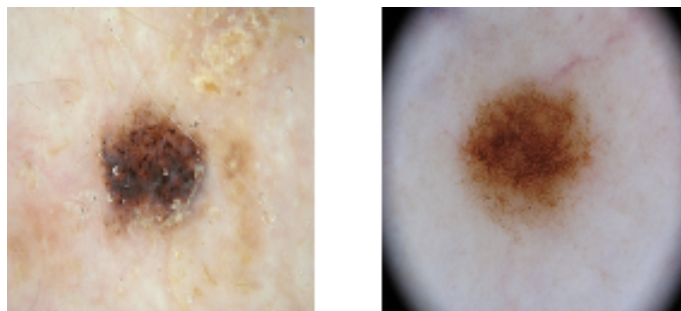


Рис. 1: Пример злокачественной и доброкачественной родинок

Далеко не все пациенты вовремя приходят за помощью, недооценивая проблему. На сегодняшний день существует множество мобильных приложений, созданных для того, чтобы сократить количество запущенных случаев. Если на прием пойдет не каждый, то уж отправить на проверку фотографию труда не составит.

Несмотря на наличие аналогов, приложение разрабатывается в связи с желанием внедрять проект MIRF<sup>3</sup> в прикладные программы. Помимо этого, к нашей кафедре обратилась крупная сеть клиник с просьбой написать подобное приложение для своих нужд.

---

<sup>1</sup>Метастаза — отдалённый вторичный очаг патологического процесса, возникший при перемещении вызывающего его начала из первичного очага болезни через ткани организма

<sup>2</sup>Рисунок взят из репозитория [5]

<sup>3</sup>Краткий обзор MIRF приведен в разделе 2

# 1 Постановка задачи

Целью работы является разработка мобильного приложения для определения рака кожи. Для её выполнения были поставлены следующие задачи:

1. Провести обзор существующих решений.
2. Разработать графический интерфейс.
3. Реализовать программную часть приложения.
4. Выбрать датасет и протестировать на нем приложение.
5. Если апробация не даст удовлетворительных результатов, написать нейронную сеть и протестировать ее.
6. Задokumentировать код.

## 2 Обзор

В данном разделе представлен обзор MIRF, обзор существующих приложений для определения рака кожи, приведен перечень используемых инструментов и обоснован их выбор.

### 2.1 MIRF

MIRF[3] - кафедральный проект, работу над которым ведут как обучающиеся, так и сотрудники университета. Это платформа с открытым исходным кодом для разработки медицинских приложений, использующих различные типы медицинских изображений.

#### 2.1.1 Архитектура

Библиотека представляет собой набор модулей для выполнения разных задач. Они делятся на пакеты Core (минимальный набор модулей, необходимых для работы) и Features (набор модулей с пользовательской функциональностью).

Для удобной работы с MIRF пользователю необходимо создавать конвейеры. Конвейер - последовательность обработчиков исходных данных. При выполнении обработчиков использовался подход Pipes and Filters [7]

Диаграмма классов пакета Core представлена на Рис. 2<sup>4</sup>

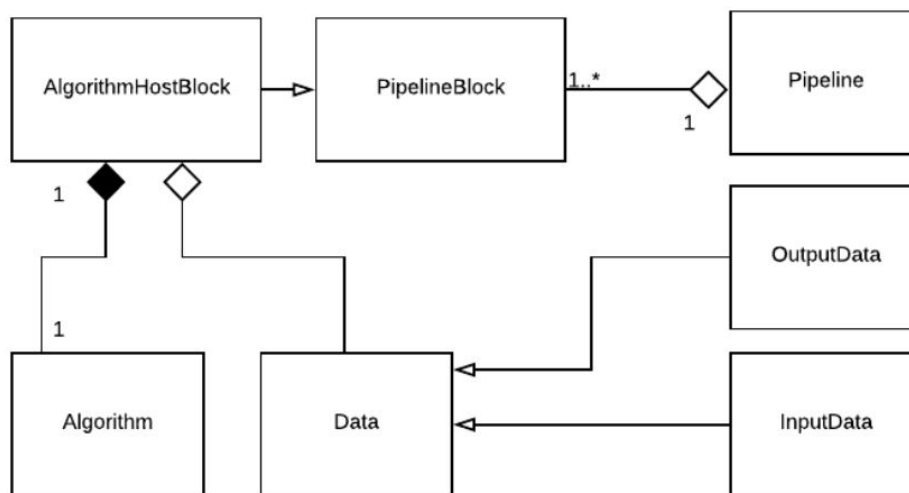


Рис. 2: Диаграмма классов пакета Core

<sup>4</sup>Рисунок взят из работы [4]

### 2.1.2 MIRFskinCancer

Приложение MIRFskinCancer было разработано в 2019 году в качестве примера использования библиотеки MIRF. Его описание можно найти в работе [4]. В 2021 году им заинтересовалась крупная сеть клиник, поэтому была поставлена задача: доработать приложение.

## 2.2 Существующие решения

Целью данного обзора является формулировка требований к MIRFskinCancer. На что было обращено внимание:

- Камера
- Связь со специалистом
- Рекомендации
- Стоимость

Аналоги:

- ПроРодинки<sup>5</sup>
- SkinVision<sup>6</sup>
- Skinive<sup>7</sup>

Камера: приложения SkinVision и Skinive оборудованы камерами с искусственным интеллектом, что позволяет сделать более точный снимок. ПроРодинки имеет свою камеру без ИИ, с помощью которой можно удобно делать снимки. MIRFskinCancer на данный момент использует камеру по умолчанию.

Связь со специалистом: ПроРодинки предоставляет возможность связаться со специалистом прямо из приложения. В остальных приложениях такой возможности нет.

Рекомендации: Все обозреваемые приложения предоставляют пользователю рекомендации

Платное/бесплатное: SkinVision и Skinive являются платными, у Skinive есть пробный период. ПроРодинки - бесплатное.

В весеннем семестре будет предпринята попытка добавить рекомендации.

---

<sup>5</sup><https://www.prorodinki.ru/>

<sup>6</sup><https://www.skinvision.com/>

<sup>7</sup><https://skinive.ru/>

## 2.3 Используемые инструменты

- Android Studio — среда разработки Android приложений
- Kotlin — язык программирования, подходящий для написания приложений под Android
- Python — язык программирования, подходящий для написания нейронных сетей
- Tensorflow — библиотека для работы с нейронными сетями
- Gradle — система автоматической сборки
- MIRF [3] — платформа с открытым исходным кодом для разработки медицинских приложений, использующих различные типы медицинских изображений
- Dokka [6] — инструмент для создания документации Kotlin

Языки Java, Kotlin, Swift, Rust поддерживают разработку мобильных приложений. Был выбран Kotlin, так как изначально работа над MIRFskinCancer велась именно на этом языке и из-за этого в качестве среды разработки была выбрана Android Studio. По этой же причине в качестве системы сборки была выбрана система Gradle.

Библиотека MIRF выбрана, потому что это кафедральный проект и его хочется внедрять в прикладные программы. Dokka поддерживает KDoc.

## 3 Ход работы

### 3.1 Разработка графического интерфейса

Изначально существовала демо версия приложения. Его графический интерфейс представлен на Рис. 3 При нажатии на кнопку СHECK выводилось сообщение с результатом исполнения алгоритма для единственного изображения, которое нельзя изменить.



Рис. 3: Начальный интерфейс

Теперь интерфейс состоит из двух окон. Конечный графический интерфейс представлен на Рис. 4. На стартовое окно приложения помещены логотип библиотеки MIRF и кнопка **START** для перехода к основному экрану. На основном экране расположены поле для изображения, кнопки **CAMERA**, **GALLERY**, **BACK**.

**CAMERA** создает намерение открыть камеру, **GALLERY** — галерею, **BACK** — возвращает пользователя на стартовое окно.

После загрузки изображения сверху появляется текст с результатом работы приложения.

### 3.2 Разработка программной части приложения

Кнопки **START** и **BACK** предназначены для переключения между стартовым и главным окнами.

При нажатии на кнопку **CAMERA** приложение запросит доступ к камере, если его нет, и запустит ее, если соответствующее разрешение получено. Если пользователь не предоставит приложению доступ к камере, то появится сообщение о том, что для работы приложения данный доступ необходим. После того, как пользователь сделает снимок, он будет



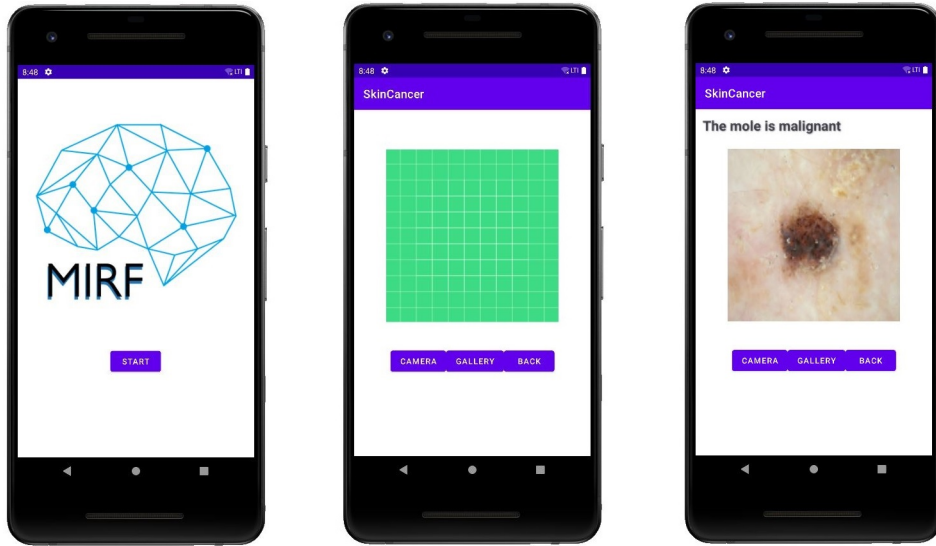


Рис. 4: Конечный интерфейс

помещен в поле для изображения и функция `detectMole()` будет запущена автоматически. Результат будет выведен на экран. Аналогично с кнопкой для загрузки изображения из галереи.

`detectMole()` создает конвейер, состоящий из двух блоков `imageReader` и `tensorflowModelRunner`. Первый принимает изображение и передает в следующий блок его представление. Второй по представлению изображения выдает результат работы запуска `TensorflowModel`. Подробнее про работу библиотеки MIRF можно прочитать в работе [4]

### 3.3 Тестирование приложения на наборе изображений

Основными критериями при выборе датасета были деление изображений родинок на доброкачественные и злокачественные и небольшой размер датасета.

Был выбран датасет Skin Cancer: Malignant vs. Benign<sup>8</sup>, как единственный подходящий. Поиск производился в Google Dataset Search по ключевым словам «skin cancer», «benign and malignant moles». Данный датасет состоит из 660 изображений родинок, 360 из которых доброкачественные, 300 — злокачественные.

Приложение прогналось на этом наборе изображений и полученные данные были занесены в таблицу, она изображена на Рис. 5

<sup>8</sup><https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign>

		Правильный ответ	
		Benign	Malignant
Результат алгоритма	Benign	118	42
	Malignant	242	258

Рис. 5: Результаты тестирования

$$A = 0.57$$

$$P = 0.52, R = 0.86, F = 0.32, \text{ где}$$

$A$  — accuracy,  $P$  — precision,  $R$  — recall,  $F$  — F-score

52% всех родинок, отмеченных приложением как злокачественные, действительно являются злокачественными, при этом получилось определить 86% всех злокачественных родинок.

### 3.4 Реализация нейронной сети

Для реализации нейронной сети были поставлены следующие задачи:

- Выбрать несколько статей с примерами кода
- Выбрать датасет для обучения и тестирования
- Обучить нейронную сеть
- Протестировать
- Внедрить ее в приложение

Статьи выбирались на сайте [paperswithcode](https://paperswithcode.com)<sup>9</sup> по поисковому запросу «skin cancer classification». Из всех были выбраны те, в которых нейронные сети были реализованы с использованием библиотеки tensorflow. После реализации нейронной сети из работы [1] было принято решение больше не рассматривать статьи, так как результат оказался приемлемым.

Для тренировки нейронной сети требуется больше изображение, так что предыдущий датасет<sup>10</sup> не подходил. Был выбран датасет melanoma<sup>11</sup>, так как в нем разделены доброкачественные и злокачественные родинки. Однако из-за нехватки вычислительной мощности пришлось использовать только 30% всех тренировочных изображений.

Нейронная сеть обучалась на протяжении пяти эпох с размером батча тридцать два. Это единственное отличие от параметров, которые были выбраны в статье. Они были изменены с целью ускорения процесса обучения. Сам процесс обучения предствален на Рис. 6

<sup>9</sup><https://paperswithcode.com>

<sup>10</sup><https://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign>

<sup>11</sup><https://www.kaggle.com/datasets/drscarlat/melanoma>

```

Epoch 1/5
75/75 [=====] - ETA: 0s - loss: 0.2953 - accuracy: 0.8825
Epoch 1: val_accuracy improved from -inf to 0.96500, saving model to weights.best.hdf5
75/75 [=====] - 747s 10s/step - loss: 0.2953 - accuracy: 0.8825 - val_loss: 0.1376 - val_accuracy: 0.9650 - lr: 1.0000e-04
Epoch 2/5
75/75 [=====] - ETA: 0s - loss: 0.1275 - accuracy: 0.9575
Epoch 2: val_accuracy improved from 0.96500 to 0.98667, saving model to weights.best.hdf5
75/75 [=====] - 838s 11s/step - loss: 0.1275 - accuracy: 0.9575 - val_loss: 0.0439 - val_accuracy: 0.9867 - lr: 1.0000e-04
Epoch 3/5
75/75 [=====] - ETA: 0s - loss: 0.1110 - accuracy: 0.9575
Epoch 3: val_accuracy did not improve from 0.98667
75/75 [=====] - 931s 12s/step - loss: 0.1110 - accuracy: 0.9575 - val_loss: 0.7860 - val_accuracy: 0.8050 - lr: 1.0000e-04
Epoch 4/5
75/75 [=====] - ETA: 0s - loss: 0.0954 - accuracy: 0.9683
Epoch 4: val_accuracy improved from 0.98667 to 1.00000, saving model to weights.best.hdf5
75/75 [=====] - 976s 13s/step - loss: 0.0954 - accuracy: 0.9683 - val_loss: 0.0090 - val_accuracy: 1.0000 - lr: 1.0000e-04
Epoch 5/5
75/75 [=====] - ETA: 0s - loss: 0.0620 - accuracy: 0.9800
Epoch 5: val_accuracy did not improve from 1.00000
75/75 [=====] - 846s 11s/step - loss: 0.0620 - accuracy: 0.9800 - val_loss: 0.0104 - val_accuracy: 0.9967 - lr: 1.0000e-04

```

Рис. 6: Процесс обучения

На тестовой выборке состоящей из 1780 доброкачественных и 1781 злокачественных родинок нейронная сеть дала результат, представленный на Рис. 7

		Правильный ответ	
		Benign	Malignant
Результат алгоритма	Benign	1779	1
	Malignant	247	1534

Рис. 7: Результаты тестирования

$$A = 0.93$$

$$P = 0.86, R = 0.99, F = 0.92, \text{ где}$$

A — accuracy, P — precision, R — recall, F — F-score

На данный момент внедрить новую нейронную сеть в приложение не вышло, так как она содержит предобученную модель ResNet50, которая пока не распознается при загрузке с помощью библиотек KotlinDL, Tensorflow-android и Deeplearning4j. В ближайшее время планируется переписать всю нейронную сеть на языке Kotlin.

## 4 Заключение

За осенний семестр был разработан первоначальный вариант приложения. За весенний спроектирована и обучена нейронная сеть для классификации родинок.

Выполненные задачи:

- Проведен обзор существующих решений.
- Разработан минимальный графический интерфейс.
- С помощью библиотеки MIRF разработана программная часть приложения.
- Спроектирована и обучена нейронная сеть
- Апробация показала, что новая нейронная сеть имеют приемлемую точность.
- Код задокументирован, документацию можно сгенерировать с помощью инструмента Dokka.
- Создано предложение изменения кода в основном репозитории MIRF<sup>12</sup>

---

<sup>12</sup><https://github.com/MathAndMedLab/MIRF2/pull/4>

## Список литературы

- [1] Convolutional neural networks for classifying melanoma images  
[https://abhinavsagar.github.io/files/skin\\_cnn.pdf](https://abhinavsagar.github.io/files/skin_cnn.pdf)  
(Дата обращение: 2022-04-14)
- [2] Всемирная организация здравоохранения  
<https://www.who.int/>  
(Дата обращения: 2021-12-12)
- [3] Главный репозиторий проекта MIRF  
<https://github.com/MathAndMedLab/MIRF2>  
(Дата обращения: 2021-12-12)
- [4] Мусатян С.А. Библиотека для создания программного обеспечения, использующего медицинские изображения. 2019.  
[https://se.math.spbu.ru/thesis/texts/Musatjan\\_Sabrina\\_Andranikovna\\_Bachelor\\_Thesis\\_2019\\_text.pdf](https://se.math.spbu.ru/thesis/texts/Musatjan_Sabrina_Andranikovna_Bachelor_Thesis_2019_text.pdf)  
(Дата обращения: 2021-12-12)
- [5] Реализация одного из алгоритмов глубокого обучения для определения рака кожи  
<https://github.com/dasoto/skincancer>  
(Дата обращения: 2021-12-12)
- [6] Репозиторий dokka  
<https://github.com/Kotlin/dokka>  
(Дата обращения: 2021-12-12)
- [7] Vermeulen Allan, Beged-dov Gabe, Thompson Patrick. The Pipeline Design Pattern. — 1995.