**6** РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В данном разделе описан как процесс установки и настройки серверной стороны программного средства колоризации изображения, так и процесс создания установочного файла клиентского приложения для платформы Android. При описании процесса установки и настройки серверной части будет описан более простой способ, использующий готовый Docker-образ. Также возможен более сложный вариант, предполагающий установку и настройку официального Docker-образа с предустановленным Caffe.

**6.1** Установка Docker

Перед выполнением дальнейших пунктов необходимо установить ПО Docker, которое можно загрузить с официального сайта разработчика.

Текущими поддерживаемыми настольными операционными системами являются:

* Mac;
* Windows.

Docker поддерживает следующие серверные ОС:

* Windows Server;
* CentOS;
* Debian;
* Fedora;
* Oracle Linux;
* RHEL;
* SLES;
* Ubuntu.

Также присутствует поддержка облачных платформ:

* AWS;
* Azure.

В данном руководстве будет описан процесс установки Docker для Windows 10.

Для установки Docker необходимо загрузить из Docker Store msi-файл и запустить его. Данное действие запустит процесс установки Docker. После установки программного обеспечения. Возможно потребуется перезагрузка операционной системы. Следует также отметить что Docker использует Hyper-V, что может повлиять на другое ПО установленное на компьютере.

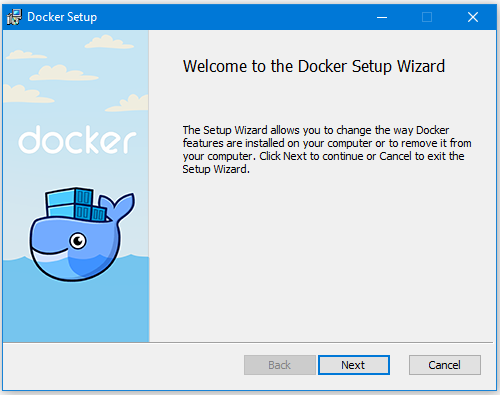


Рисунок 6.1 – Окно установки ПО Docker

После установки Docker логотип этого программного обеспечения появится в системном трее (см. рисунок 6.2).

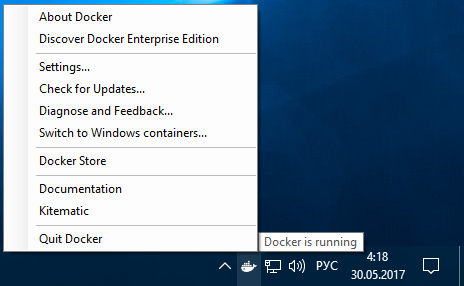


Рисунок 6.2 – Логотип Docker в системном трее

Для верификации установки необходимо выполнить команду docker version с помощью CLI-интерфейса системы.

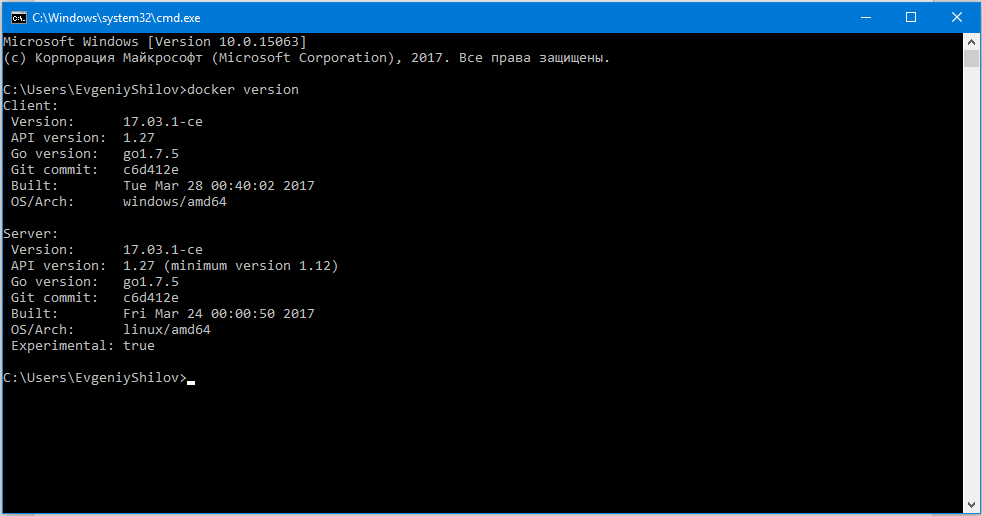


Рисунок 6.3 – Пример выполнения команды docker version

**6.2** Установка сервера из сконфигурированного образа.

В данном подразделе описана установка и настройка серверной части из подготовленного Docker-образа. Для этого необходимо выполнить команду docker pull, указав адрес Docker Hub репозитория.

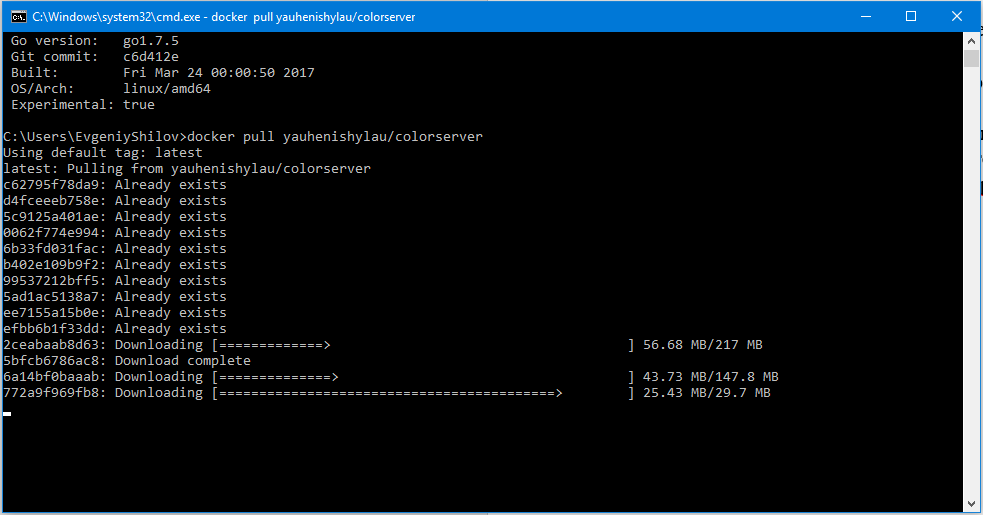


Рисунок 6.4 – Пример выполнения команды docker pull

В качестве адреса удалённого репозитория необходимо указать yauhenishylau/colorserver. Данная команда загрузит образ, содержащий все необходимые исходные файлы. После загрузки образа, перед созданием контейнера необходимо узнать, адрес IP-адрес интерфейса, по которому клиентское приложение будет получать доступ к серверу. Для этого в ОС Windows необходимо выполнить команду ipconfig (см рисунок 6.5).

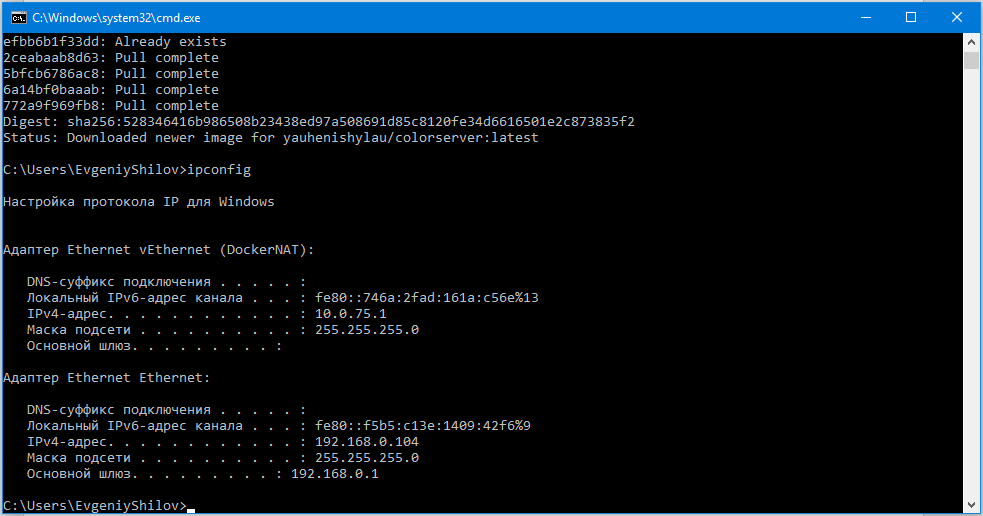


Рисунок 6.5 – Пример выполнения команды ipconfig

Следующим шагом является создание Docker-контейнера при помощи команды docker run. При этом необходимо добавить флаг –p для того, чтобы связать порт ОС-хоста с внутренним портом контейнера.

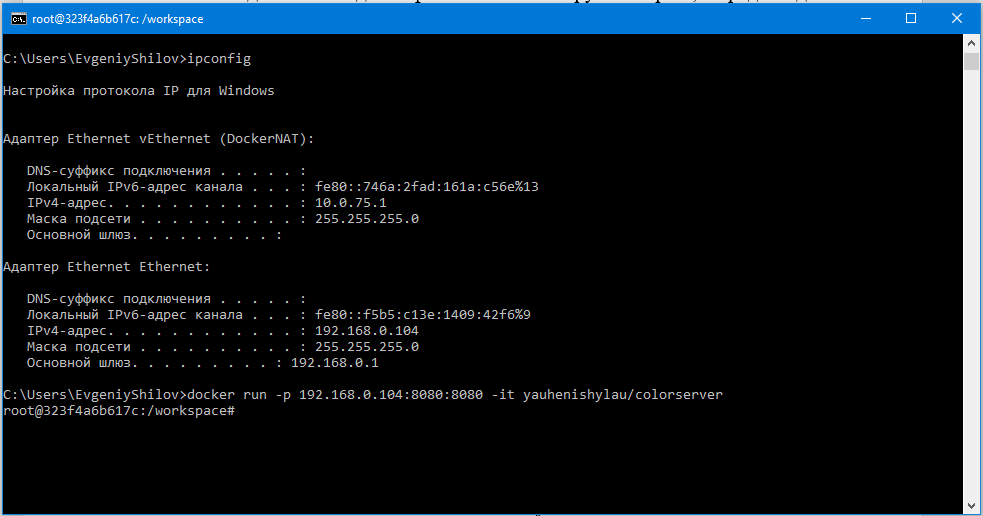


Рисунок 6.6 – Создание контейнера Docker

После создания контейнера необходимо запустить файл server.py (см. рисунок 6.7).

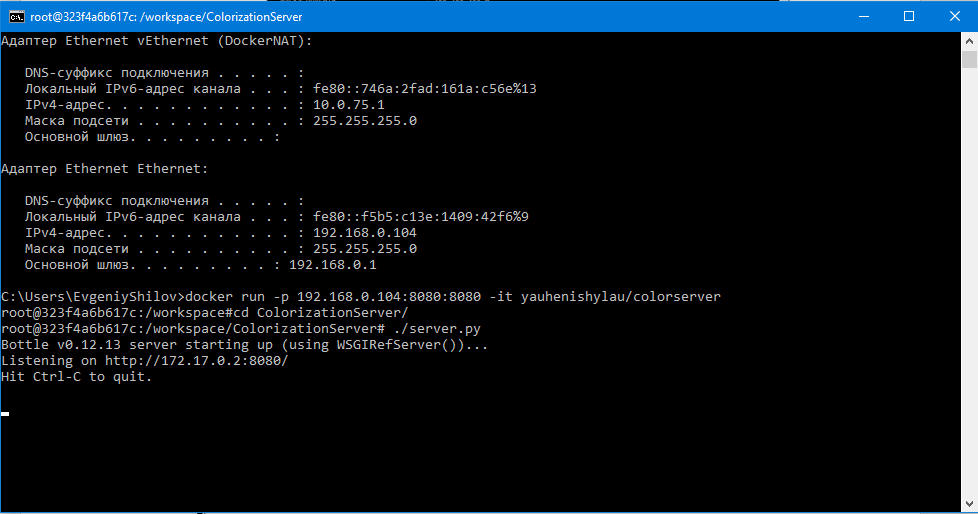


Рисунок 6.7 – Запуск сервера

После выполнения данной команды сервер готов к использованию. Доступ к серверу можно получить по паре адрес-порт указанному при создании контейнера. При завершении работы контейнера, его можно в дальнейшем перезапустить при помощи команды docker start, указав при этом имя созданного контейнера. Для того, чтобы узнать имя контейнера необходимо выполнить команду docker container ls (см рисунок 6.8).

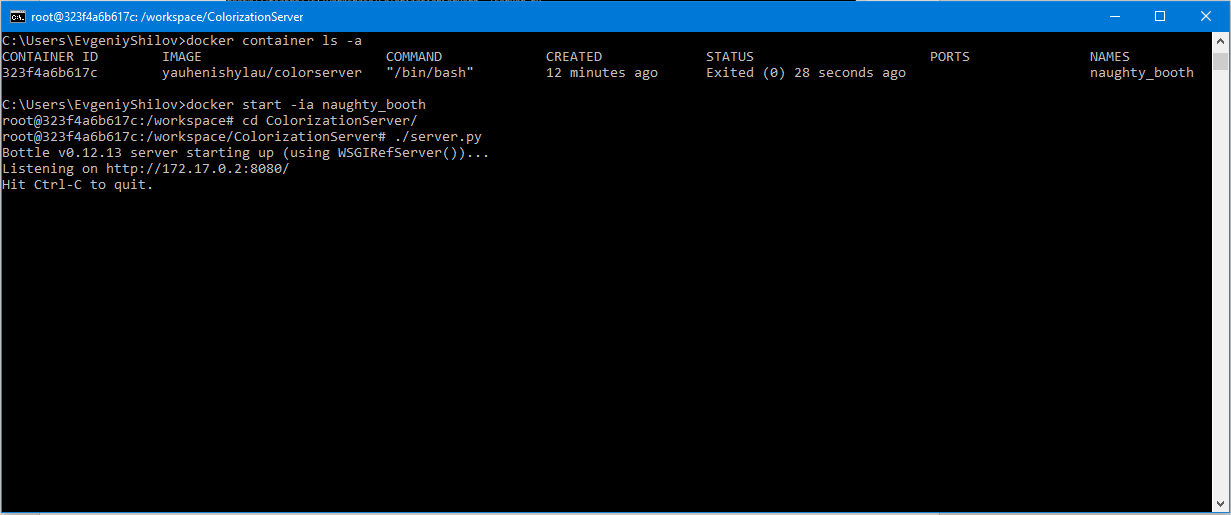


Рисунок 6.8 – Перезапуск сервера

**6.3** Создание установочного файла для платформы Android

Для выполнения данного шага необходимо установить следующее ПО:

* Java Developer Kit – ПО компании Oracle, необходимое для разработки программ для JVM;
* Android Studio – IDE для создания приложений для платформы Android;
* Android SDK – комплект средств разработки, устанавливаемый вместе со средой разработки Android Studio.

Для создания APK-файла необходимо запустить Android Studio и выбрать пункт «Open an existing Android Studio project». Затем необходимо указать путь к каталогу с исходным кодом проекта (см рисунок 6.9). Исходный код проекта содержится в архиве на носителе, прикреплённом к данному дипломному проекту.

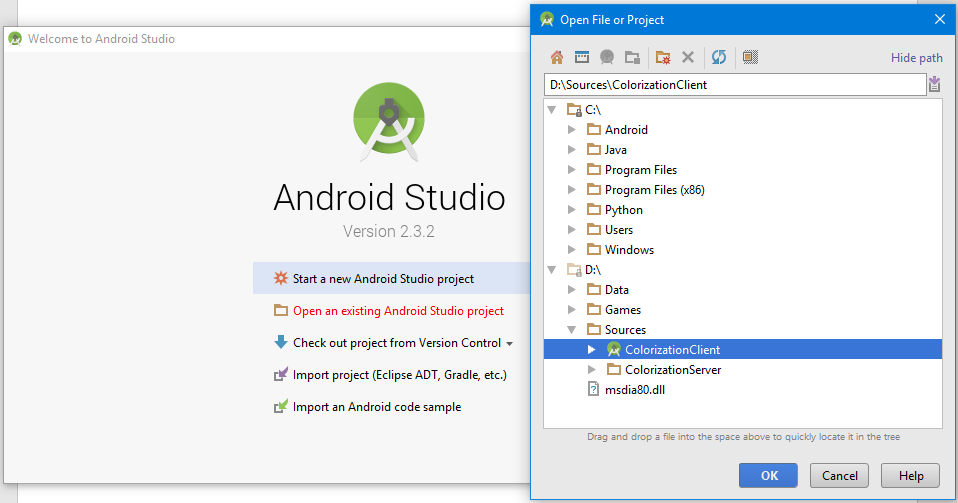


Рисунок 6.9 – Открытие проекта Android Studio

После открытия проекта подобным способом система сборки Gradle сгенерирует недостающие конфигурационные файлы. После этого, как Gradle построит необходимые конфигурационные файлы, необходимо указать адрес сервера в классе API. Данный класс размещён в пакете api, который должен быть размещён в каталоге проекта Android Studio, который размещён по пути srс\main\java\by\evgeniyshilov\colorizationclient\api.

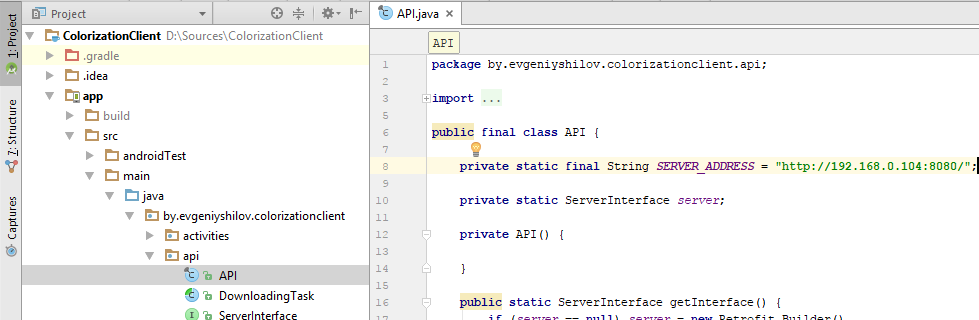


Рисунок 6.10 – Расположение файла API.java

В данном файле для указания адреса сервера используется строка SERVER\_ADDRESS. После изменения адреса сервера можно приступить к сборке APK-файла. Для этого необходимо выбрать пункт меню Build и в появившемся контекстном меню выбрать пункт Build APK.

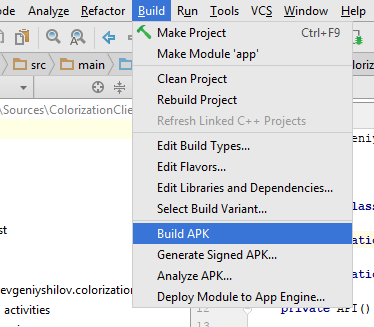


Рисунок 6.11 – Выбор пункта меню для построения APK-файла

Данное действие приведёт к построению APK файла. После завершения пользователь получит уведомление об успешном окончании генерации, а сам файл появится в каталоге app\build\outputs\apk проекта.

**6.4** Руководство по использованию Android-приложения

После установки APK-файла на устройство, необходимо запустить приложение из основного меню системы.

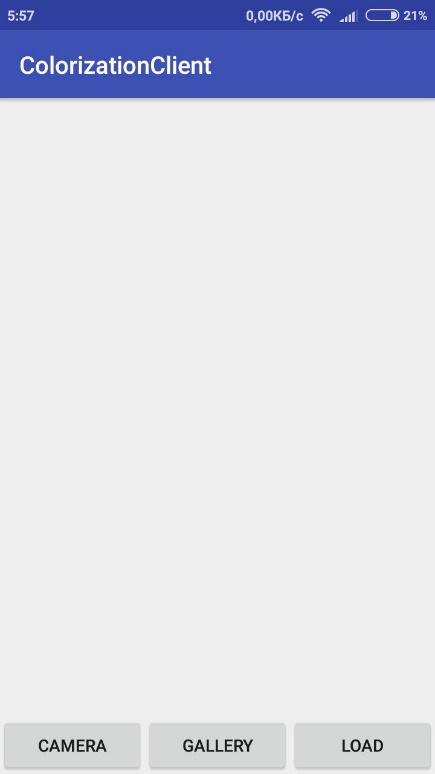


Рисунок 6.12 – Экран для выбора изображения

На данном экране пользователю доступны три кнопки (см. рисунок 6.12). Кнопка «LOAD» не приводит к выполнению какого-либо действия, пока пользователь не выберет изображение.

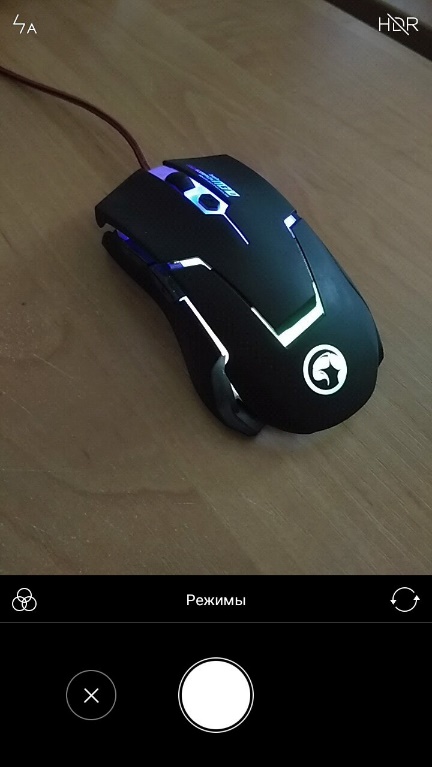


Рисунок 6.13 – Получение изображения с помощью камеры устройства.

После того, как пользователь выберет изображение, его полутоновая версия будет отображена на экране (см. рисунок 6.14)



Рисунок 6.14 – Просмотр полутоновой версии выбранного изображения

Если пользователь сочтёт нужным выбрать другое изображение, то он может снова воспользоваться кнопками «CAMERA» и «GALLERY». В противном случае пользователь должен нажать кнопку «LOAD» для начала колоризации.



Рисунок 6.15 – Экран ожидания результата колоризации

После окончания колоризации индикатор загрузки будет скрыт, а на экране будет отображено обработанное изображение (см рисунок 6.16).



Рисунок 6.16 – Отображение результата колоризации

Для колоризации другого изображения пользователю необходимо нажать на кнопку «BACK» или соответствующий контроллер своего устройства. Для выхода из приложения пользователю необходимо повторно нажать на клавишу «BACK» своего устройства.