**3** ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

В данном разделе детально рассматривается, как функционирует программа. Раздел состоит из подразделов, каждый из которых описывает функционирование отдельного модуля программы, а также интерфейс взаимодействия с другими модулями приложения. Сами функциональные блоки также будут различаться между собой по месту, где они будут реализованы.

Стоит заметить, что функциональное проектирование модулей и описание интерфейсов взаимодействия будет зависеть от языка программирования и в некоторых случаях от используемого фреймворка, что объясняет некоторое различие подразделов данной записки.

**3.1** Структура модуля колоризации

Как было заявлено выше, для построения программного средства колоризации полутоновых изображений необходимо реализовать нейронную сеть, которая и будет обрабатывать изображения. Так как для реализации нейронной сети был выбран фреймворк Caffe, в пояснительной записке будет кратко изложены основные принципы построения нейросетей в данном фреймворке.

В библиотеке Caffe топология нейросетей, исходные данные и способ обучения задаются с помощью конфигурационных файлов в формате prototxt. Топология сетей описывается с помощью слоёв, каждый из которых имеет свой тип, параметры и связь с другими слоями нейронной сети. Ниже приведён пример описания свёрточного слоя в prototxt-файле:

# Объявление слоя

layer {

# Наименование слоя

name: "conv3\_1"

# Тип слоя

type: "Convolution"

# Ссылка на слой, за

# которым необходимо

# создать текущий

bottom: "conv2\_2norm"

top: "conv3\_1"

# Параметры слоя

convolution\_param {

# Размер выхода слоя

num\_output: 256

pad: 1

# Размер ядра свёртки

kernel\_size: 3

}

}

Файл с расширением prototxt для описания процесса обучения в сообществе пользователей Caffe принято называть «solver». Подобные файлы содержат набор обязательных и опциональных параметров для процесса обучения. Таковыми параметрами являются:

* путь к файлу с конфигурацией сети;
* периодичность тестирования во время обучения;
* параметры стохастического градиентного спуска;
* максимальное количество итераций;
* архитектура, на которой будут проводиться вычисления;
* путь для сохранения обученной сети.

Пример solver-файла будет рассмотрен в дальнейшем вместе с указанием параметров для обучения нейросети.

Для запуска обучения свёрточной нейронной сети и обработки входных данных будут использоваться соответственно интерфейс командной строки и Python-интерфейс, описание функционала, которого мы также рассмотрим в дальнейшем.

**3.2** Архитектура нейронной сети

Свёрточная нейронная сеть – специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном и нацеленная на эффективное распознавание изображений. Основной идеей свёрточной нейронной сети является чередование слоёв, выполняющих операцию свёртки со слоями субдискретизации. По определению, свёртка – это математическая операция, применённая к двум функциям, порождающая третью функцию, которая иногда может рассматриваться как модифицированная версия одной из первоначальных. В контексте обработки изображений рассматривается понятие ядра свёртки – матрицы фиксированного размера, содержащей коэффициенты, значения которых перемножаются со значениями каналов пикселей изображения. Данная маска размещается в углу изображения и высчитывается значение свёртки для пикселя в центре маски. Затем маска сдвигается к следующему пикселю, и операция повторяется.

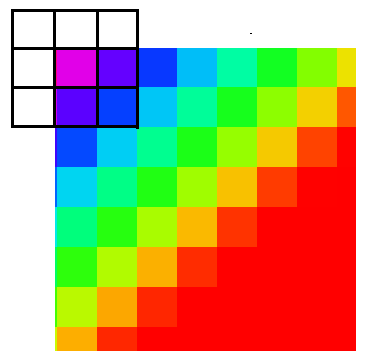


Рисунок 3.1 – Итерация обработки изображения свёрточным фильтром

Слои субдискретизации используются для уменьшения размера, последующих слоёв. Обычно из группы соседних результатов предыдущего слоя свёртки выбирается максимальный, который и становится входом для следуюшего слоя. Вставка данных слоёв увеличивает производительность нейронной сети, при этом незначительно влияя на результат.

Каждый свёрточный слой имеет несколько фильтров, после обработки которыми образуются карты признаков. Так при углублении в сеть размер слоя уменьшается, а количество карт признаков увеличивается. При этом теряется информация о расположении более мелких признаков на изображении (линий, окружностей), и появляется информация о наличии более крупных признаков на изображении (сложных фигур, объектов). Затем следуют несколько полносвязных слоёв, которые обрабатывают информацию о признаках. Подобные сети часто выполняют задачу классификации изображений. Примером такой архитектуры является VGG Convolutional Neural Network.

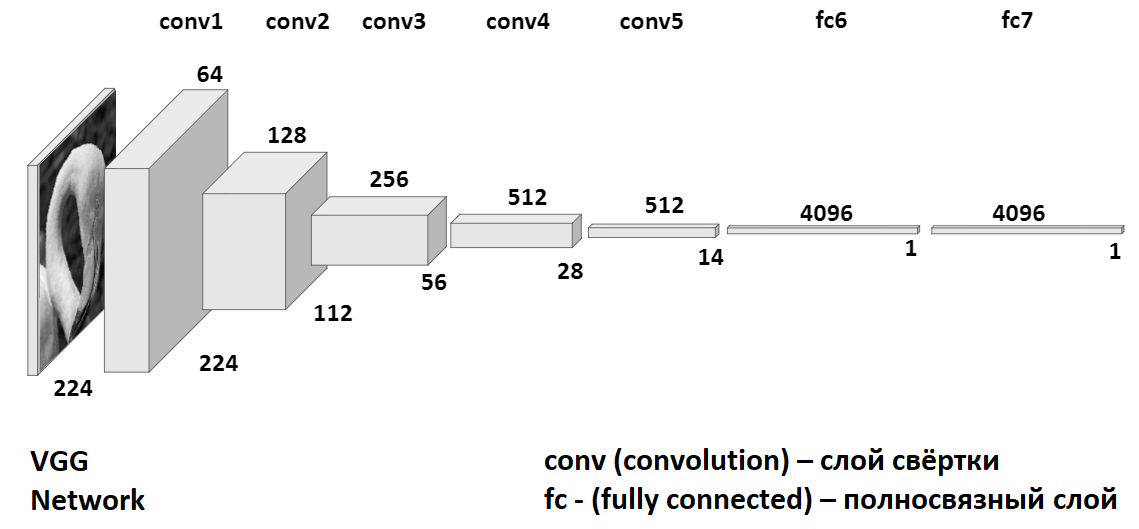


Рисунок 3.2 – Архитектура VGG сети

Для решения задачи колоризации данная архитектура не подходит из-за наличия полносвязных слоёв, при использовании которых теряется о расположении признаков на изображении теряется полностью. Также количество нейронов в последнем слое нейронной сети должно превышать или хотя бы быть равным количеству нейронов входного слоя. Исходя из этого следует, что текущую архитектуру необходимо модифицировать под задачу колоризации.

Для начала необходимо определить формат выходных данных. Для этого будет сделан выбор цветовой модели, в которой будет представлено обработанное изображение. В первую очередь модель должна быть распространённой для упрощения процесса перевода изображений из RGB и обратно. Также выбранная модель должна охватывать достаточно большое цветовое пространство.

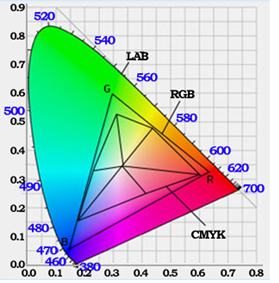


Рисунок 3.3 – Сравнение покрытия цветового пространства разными цветовыми моделями

Подходящей цветовой моделью является LAB. В данной цветовой модели цвет представлен двумя каналами: A и B. Яркость представлена третьим каналом L. В первую очередь данная модель охватывает более широкий диапазон цветового спектра, чем другие цветовые модели (см. рисунок 3.3). Также важным достоинством этой модели является наличие отдельного канала для яркости. При подаче канала яркости L на вход нейросети, необходимо лишь получить значения каналов A и B.

Следующим этапом является квантование LAB пространства, которое покрывается RGB моделью (см рисунок 3.4). Для этого делим данное пространство на квадраты со стороной 10. Количество подобных квадратов будет равным 313. Каждый из квадратов соответствует одному цвету, который является средним для всех значений цветов в квадрате.

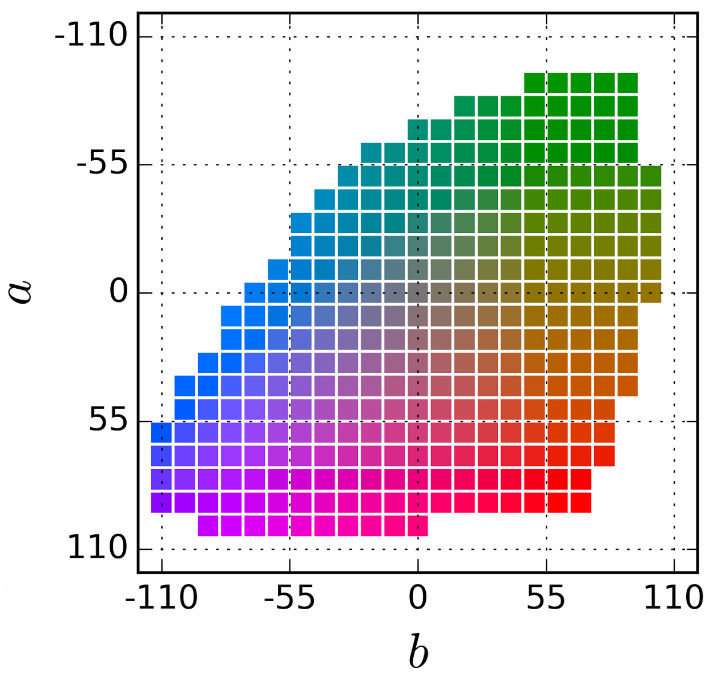


Рисунок 3.4 – Квантование цветового пространства LAB

На основании полученных цветов можно построить слой из 313 карт признаков, каждая из которых будет содержать вероятность того, что группы пикселей будут принимать цвет соответствующей этой карте. Входом для данного слоя будет являться результат последнего слоя свёртки, а на основании результатов группе пикселей присваиваться цвет наибольшей вероятностью (см. рисунок 3.5).

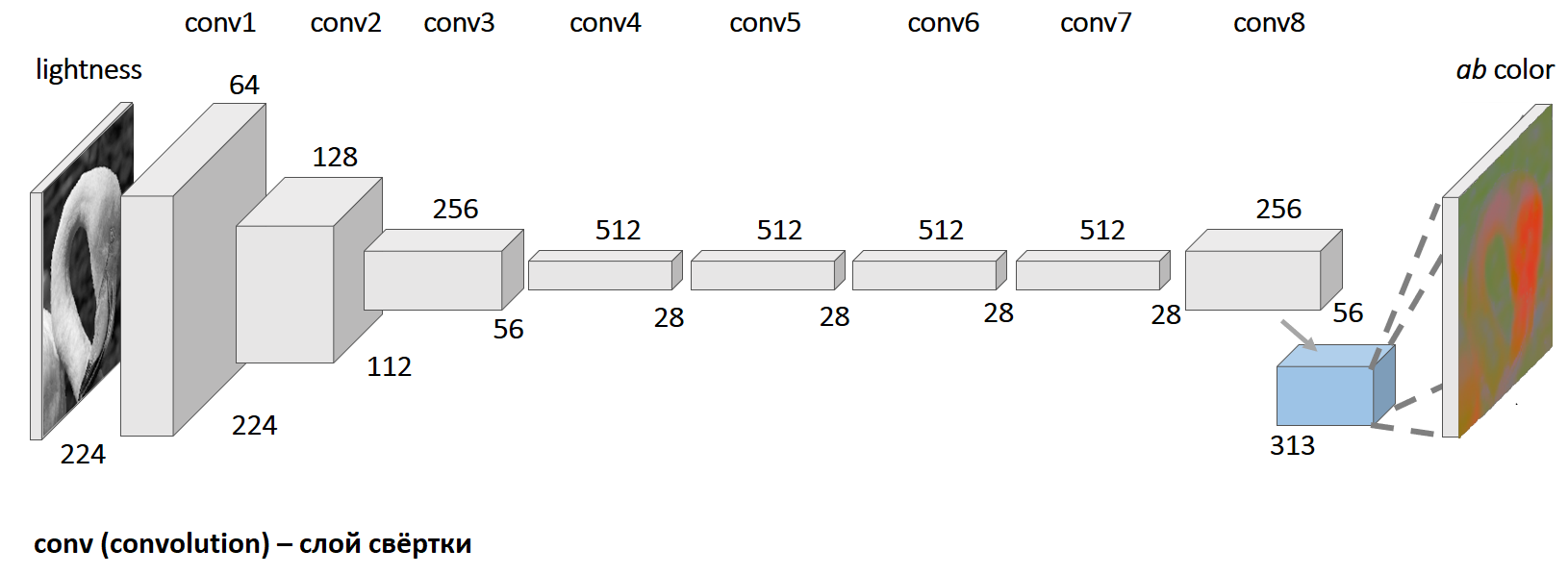


Рисунок 3.5 – Архитектура нейросети для колоризации изображения

Стоит также отметить, что полносвязные слои будут заменены свёрточными слоями с некоторыми модификациями. Реализация этих слоёв будет описана в следующих разделах, а также в приложенном листинге.