Литература

1 А.Мерибан Communication without words[Текст]/А. Мерибан. – 1968. – С. 53-56.

2 Б. Фасель, Д. Лютин, Automatic Facial Expression Analysis: A Survey, Pattern Recognition[Текст]/ Б. Фасель, Д. Лютин. – 2003. - С. 259-275.

3 К. Рао,Т. Кумар, К. Ануша, Emotion Recognition from Speech – 2012. - (https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.437.2775&rep=rep1&type=pdf)

4 И. Саксмит., С. Алиссон, С. Барон-Коэн, Empathy and emotion recognition in people with autism, first-degree relatives, and controls[Текст]/ И. Саксмит., С. Алиссон, С. Барон-Коэн – 2013. – С. 98-105.

А.Д. Божимов, О.П. Солдатова

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ СЖАТИЯ МОДЕЛИ НЕЙРОННОЙ СЕТИ ДЛЯ ПЕРЕДАЧИ ПРОИЗВОЛЬНОГО СТИЛЯ ИЗОБРАЖЕНИЯ

(Самарский университет)

Современные реалии всё чаще подталкивают нас к использованию нейронных сетей при решении обширного диапазона задач. Одним из направлений, где реализация алгоритма средствами классического программирования является труднореализуемой или и вовсе невыполнимой, является художественное творчество. Развитие технологий машинного обучения позволило появиться успешным попыткам воссоздания уникальных особенностей художественных произведений техническими средствами. Это обусловлено в первую очередь способностью моделей к «обучению» - процессу, где нейронная сеть должна выявлять сложные зависимости между входными и выходными данными [1].

Большинство существующих алгоритмов переноса стиля изображения требуют отдельного обучения сети и создания новой модели для работы с каждым новым стилем. Это накладывает серьезные ограничения на используемое аппаратное обеспечение, а также увеличивает затраты времени [2]. Оптимальным решением было бы иметь модель, которая способна выполнять быструю передачу стиля на любых парах изображения, не требуя переобучения.

Рассматриваемый процесс переноса произвольного стиля изображения состоит из двух этапов: получение вектора стиля из одного изображения и стилизация другого на основе векторизованных данных.

Для выделения стиля на первом этапе используется сеть предсказания, создающая на выходе 100-мерный вектор. Такой подход позволяет также комбинировать стили, используя средневзвешенное значение [3].

На втором этапе стилизации применяется сеть переноса стиля, принимающая чистое изображение и вектор представления стиля и на выходе создаю-

щая стилизованное изображение. Такой подход позволяет устанавливать «глубину проникновения» стиля в исходное изображение.

Была разработана программная система на базе библиотеки машинного обучения TensorFlow.js. Язык JavaScript выбран для обеспечения кроссплатформенности и высокой повторяемости результатов. Используемые предобученные модели были взяты из общедоступного репозитория Google Magenta [4] и переведены для TensorFlow.js при помощи TensorFlow.js converter [5].

Благодаря возможности TensorFlow.js использовать WebGL — кроссплатформенный API для графического процессора на основе OpenGL ES 2.0, для ускорения, процесс переноса стиля выполняется на современных устройствах, включая смартфоны, в течение нескольких секунд. Однако, учитывая кроссплатформенность решения, стоит обратить внимание на объем загружаемых моделей. В конкретном случае это 36.3 МБ для сети предсказания стиля, основанной на Inception v3, и 7.9 МБ для сети переноса. Их суммарный объем составляет 44.2 МБ, что довольно много при работе с, например, web-приложением.

Для сжатия модели предсказания был использован метод «дистилляции» [6]. Суть метода заключается в обучении малой нейронной сети для повторения выходных значений большой нейронной сетью. При «дистилляции» была использована модель MobileNetV2 по нескольким причинам:

- модели MobileNetV2 показывают хорошие результаты при «дистилляции»;
- имеют качественную реализацию с открытым исходным кодом;
- имеют версию для целевого TensorFlow.js.

Изображение стиля пропускается через обе сети. Среднеквадратичное отклонение двух полученных представлений стиля используется в качестве значения потерь для обновления весов малой сети. Данный алгоритм представлен на рисунке 1.

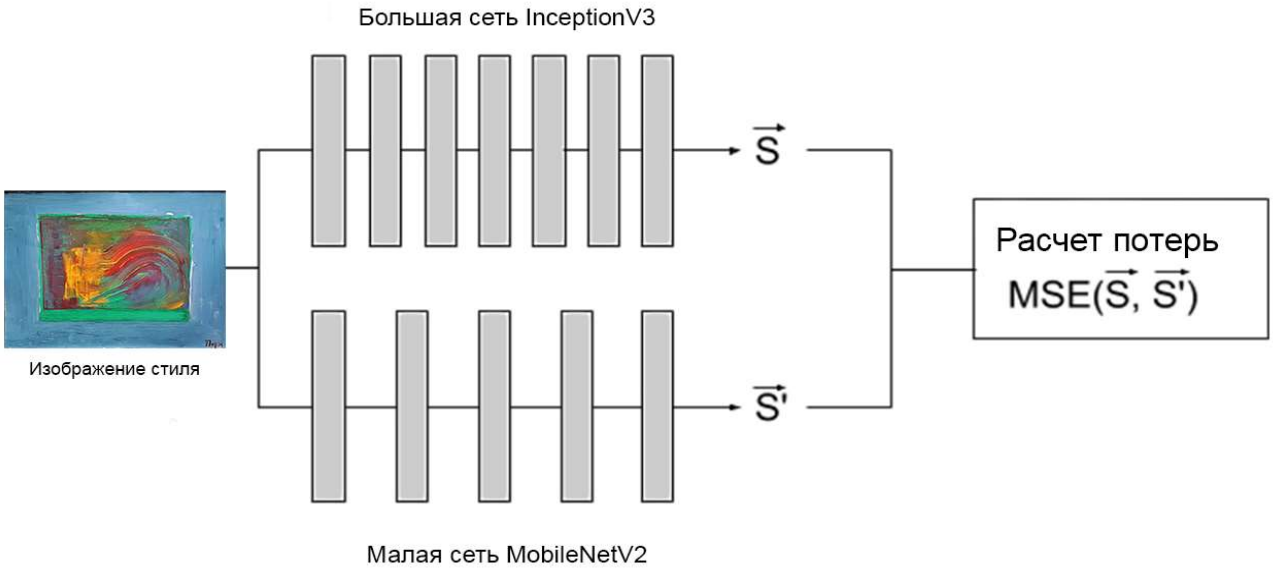

Рисунок 1 – Процесс обучения

Результаты, полученные после сжатия, представлены в таблице 1.

Таблица 1 – Среднее время работы для модели предсказания

	Объем модели, МБ	Среднее время работы, миллисекунды
До сжатия	36.3	4972
После сжатия	9.6	1013

После завершения обучения размер полученной сети уменьшился в 3,8 раза, а время выполнения в среднем в 4,9 раза.

Сравнить результаты работы сетей до и после сжатия можно на рисунке 2.

Рисунок 2 – Сравнение результатов работы сетей

При изучении полученных результатов выяснилось, что после сжатия большую часть времени работы системы (в среднем 1138 миллисекунд) занимает процесс переноса стиля, а значит, для ускорения работы в дальнейшем необходимо произвести оптимизацию и сжатие модели переноса стиля, например, переведя ее на свертку с разделением по глубине (Depthwise Separable Convolutions) [7].

По завершении процедуры сжатия удалось добиться значительного снижения суммарного объема моделей с 44.2 МБ до 17.5 МБ, при этом сохранив функционирование, хоть и с небольшим отклонением от результатов исходной модели, а также добиться многократного прироста в скорости.

Литература

1. Обучение нейронной сети [Электронный ресурс]. URL: https://neuronus.com/theory/nn/238-obucheniya-nejronnoi-seti.html (дата обращения: 14.04.2021).

2. Нейронные сети и глубокое обучение. Учебный курс [Текст]/ Аггарвал Чару – М.: ООО “И. Д. Вильямс”. 2020. – 752 с.

3. Exploring the structure of a real-time, arbitrary neural artistic stylization network [Электронный ресурс]. URL: https://arxiv.org/abs/1705.06830 (дата обращения: 02.04.2021)

4. TensorFlow Hub – Magenta – arbitrary image stylization [Электронный ресурс]. URL: https://tfhub.dev/google/lite-model/magenta/arbitrary-image-stylization-v1-256/fp16/prediction/1 (дата обращения: 02.04.2021)

5. tensorflow/tfjs: A WebGL accelerated JavaScript library for training and deploying ML models. [Электронный ресурс]. URL: https://github.com/tensorflow/tfjs (дата обращения: 02.04.2021)

6. Distilling the Knowledge in a Neural Network [Электронный ресурс]. URL: https://arxiv.org/abs/1503.02531 (дата обращения: 02.04.2021)

7. Depthwise separable convolutions for machine learning [Электронный ресурс]. URL: https://eli.thegreenplace.net/2018/depthwise-separable-convolutions-for-machine-learning/ (дата обращения: 15.04.2021)

Л.В. Болотникова, И.В. Лёзина

АВТОМАТИЗИРОВАННАЯ СИСТЕМА РАСПОЗНАВАНИЯ РИМСКИХ ЦИФР ПРИ ПОМОЩИ НЕЙРОННОЙ СЕТИ КОХОНЕНА.

(Самарский университет)

Распознавание образов — научная дисциплина, целью которой является выявление объектов по нескольким критериям или классам. Теория распознавания объектов представляет собой раздел информатики, который основывается на разработке основ и методов идентификации предметов, явлений и сигналов. Потребность в таком распознавании возникает во многих областях, начиная с машинного зрения, символьного распознавания, диагностики в медицине, распознавания речи и заканчивая узко специальными задачами [1].

Иногда подобное распознавание требуется для восстановления текстового документа. К примеру, для поврежденных литературных текстов часто требуется восстановить последовательность повествования для получения наиболее точного восприятия информации. Для этого необходимо распознавать римские цифры, которые часто используются в подобных текстах для нумерации глав, заголовков и страниц.

Для распознавания образов римских цифр была разработана автоматизированная система, основанная на нейронной сети Кохонена.

Она представляет собой двухслойную сеть, где каждый нейрон первого (распределительного) слоя соединен со всеми нейронами второго (выходного) слоя, которые расположены в виде двумерной решетки.

Нейроны выходного слоя называются кластерными элементами, их количество определят максимальное количество групп, на которые система может разделить входные данные. Увеличивая количество нейронов второго слоя можно увеличивать детализацию результатов процесса кластеризации.

Для обучения сети Кохонена используется соревновательный метод [2]. На каждом шаге обучения из исходного набора данных случайно выбирается один вектор. Затем производится поиск нейрона выходного слоя, для которого