МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное   
образовательное учреждение высшего образования

«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики  
Кафедра суперкомпьютеров и общей информатики

**Отчет по лабораторной работе №2**

Дисциплина: «Облачные и высокопроизводительные вычисления»

Тема: **«Использование GPU при глубоком обучении для семантической сегментации изображений»**

Выполнил: Печенин В.А.

Группа: 6133-010402D

Самара 2020

**ЗАДАНИЕ**

Задача: реализовать алгоритм для обучения и тестирования семантической сегментации

Язык: Python

Входные данные: обучающие и тестовые матрицы (1000 и 50 штук по 6 проекций) входных изображений размером 480х480х3 и размеченных изображений размером 440х440 или 440х440х5.

Выходные данные: проверка точности семантической сегментации + время обучения нейронных сетей.

Отчет о проделанной лабораторной работе - это git-репозиторий с исходным кодом реализации + описание проделанной работы там же в readme.

**ХОД РАБОТЫ**

Компьютер, на котором выполнялась лабораторная работа, имеет: CPU: Intel(R) Core(TM) i5-3470 @3.2 GHz, 4 ядра; GPU: NVIDIA GeForce GTX 1050 Ti .

1. Общий ход решения и объект исследования

Объектом исследования выступаю проекции с stl-моделей сектора из трех лопаток соплового аппарата турбины (рисунок 1).

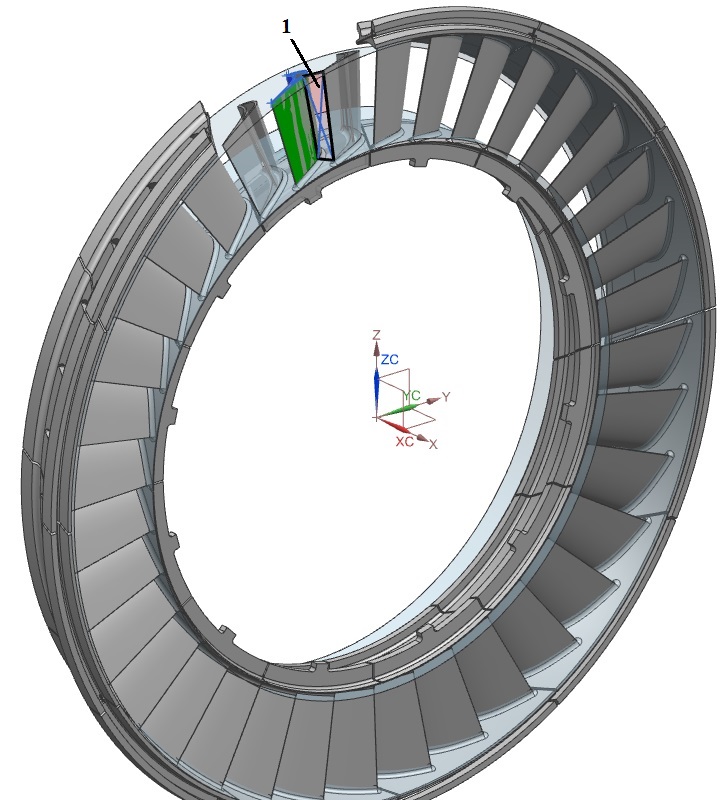


Рисунок 1 – Модель сборки соплового аппарата турбины

После измерения производится сегментация объектов на отдельные грани, основанная на вычислении и использовании тензора главных кривизн в вершинах stl моделей были использованы результаты работы [1].

На рисунке 2 приведены результаты работы алгоритмасегментации граней stl.

|  |  |
| --- | --- |
| а | Сегментация после алгоритма  б |

Рисунок 2 – stl-модель на входе (а) и после сегментации (б)

Распознавание сегментированных граней нужно для автоматического выбора требуемых для дальнейших расчетов геометрических параметров (площадь проходного сечения) граней. Распознавание граней связано с использованием свёрточных нейронных сетей. Для обучения нейронной сети необходимо подготовить обучающую выборку, а для оценки качества – тестовую. Входные данные в нейронные сети представляют собой данные с проекций детали на шесть плоскостей (вид спереди, сзади, слева, справа, сверху и снизу), параллельных координатным плоскостям. Объект помещён как бы в куб, грани которого параллельны плоскостям координат. Для подготовки проекций был использован алгоритм Робертса.

Выходными данными так же являются шесть проекций деталей, с той разницей, что грани на них размечены. Целевыми гранями при распознании являются грани, по которым производится сопряжение сборки и контроль. Соответственно эти грани выделяются разными цветами, остальные грани тела имеют единый и отличающийся цвет. При распознавании каждого вида объектов используются отдельные обученные нейронные сети.

1. Выбор архитектуры и разработка нейросетевой модели для задачи распознавания геометрических элементов.

Как отмечалось выше, распознавание граней связано с использованием свёрточных нейронных сетей. Для обучения нейронной сети необходимо подготовить обучающую выборку, а для оценки качества – тестовую.

Важной задачей является выбор архитектуры нейронной сети для выполнения задачи распознавания геометрических элементов. Для выполнения распознавания были использованы архитектуры U-net [2], SegNet [3] и Enet [4].

Архитектура сети U-Net приведена на рисунке 3.

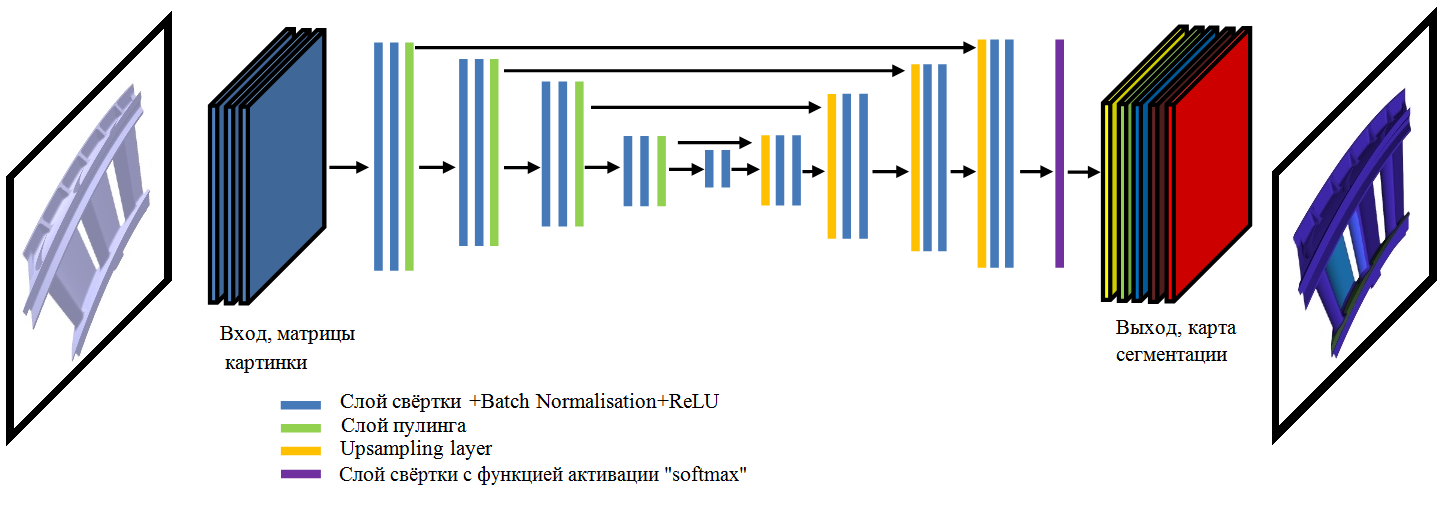


Рисунок 3 – Архитектура сети U-net

Для окончательной сегментации сегментированные картинки соотносятся с фасетами stl. Шесть сегментированных картинок-проекций находятся на гранях «куба», в который заключена stl-модель и последовательно (начиная с первой проекции), сегментированные пиксели в пространстве соотносятся с ближайшими к ним фасетами. Соответственно выбирается сегмент, имеющий наибольшее пересечение с массивом соотнесенных с распознанной проекцией фасет.

1. Экспериментальная часть

С использованием оптического 3D сканера RANGEVISION Pro2M было выполнено измерение партии из шести секторов лоптаток турбин (рисунок 4).

|  |  |
| --- | --- |
| а) | б) |

Рисунок 4 – а) комплекты секторов лопаток турбины б) сканирование сектора

Количество фасет каждой stl после сканирования имеет тот же порядок (около 420000), что и STL, созданная по теоретической модели, как отмечалось в предыдущем разделе. STL по теоретической модели содержит 205760 вершин, объединенных в 411524 фасеты (число строк в матрице **F**m×3). С использованием разработанного алгоритма для решения задачи распознавания геометрических элементов в измеренном массиве точек были измеренные данные были сегментированы на области и распознаны элементы на персональном компьютере с процессором AMD Ryzen 7 2700 Eight-Core Processor, с тактовой частотой 3,2 ГГц, имеющим 8 ядер (16 логических процессоров) и графическим процессором NVIDIA GeForce GTX 1080 Ti. Объем ОЗУ 32 Гб. Алгоритмы распознавания с помощью нейронных сетей были реализованы в программном коде на языке Python с использованием программных библиотек TensorFlow и Keras. Для рассматриваемых архитектур нейронных сетей были написаны программы: *U-Net\_for\_Keras\_universal.py* и *support\_function\_for\_Unet.py* для U-net; *SegNet\_for\_Keras\_univeresal.py*; *support\_function\_for\_SegNet.py* и *SegNet\_example2.py* для SegNet; *Enet\_for\_Keras\_universal.py*; *support\_functions\_for\_enet.py* и *enet\_model.py* для Enet;

* 1. Cоздание посредством моделирования тестовых и обучающих наборов действительных 3D моделей, основанных на экспериментальных данных.

Твердотельная конструкторская модель сектора трех лопаток соплового аппарата турбины (рисунок 1) была преобразована в формат stl, как отмечалось в предыдущем разделе. Затем было выполнено искусственное моделирование, посредством перемещения, вращения и изменения масштаба обучающей и тестовой выборки stl-объектов секций сопловых аппаратов. Обучающая выборка 1000 случаев, тестовая 50. Для каждого экземпляра выборок (пара входные данные и выходные данные, в которые требуется преобразовать входные) было выполнено сохранение шести картинок-проекций (рисунок 5, а). Выходные данные представляют собой проекции размеченных stl, где разными цветами обозначены целевые грани для распознавания (рисунок 5, б).

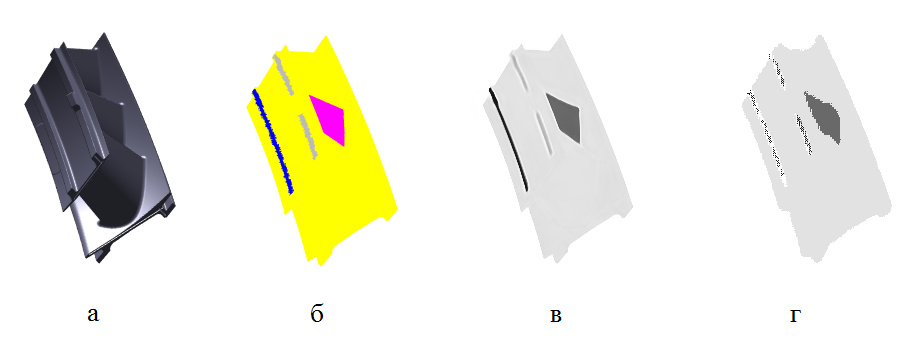


Рисунок 5 – а) исходная проекция stl; б) размеченная проекция; распознавание с использованием в) U-net; г) Enet

Были обучены нейронные сети трех архитектур (U-net, SegNet и Enet), и выполнено распознавание тестовой выборки. На рисунке 5 приведены результаты распознавания одной из проекций с помощью U-net и Enet. На рисунках 6-8 приведены скрины процесса обучения рассматриваемых нейронных сетей.

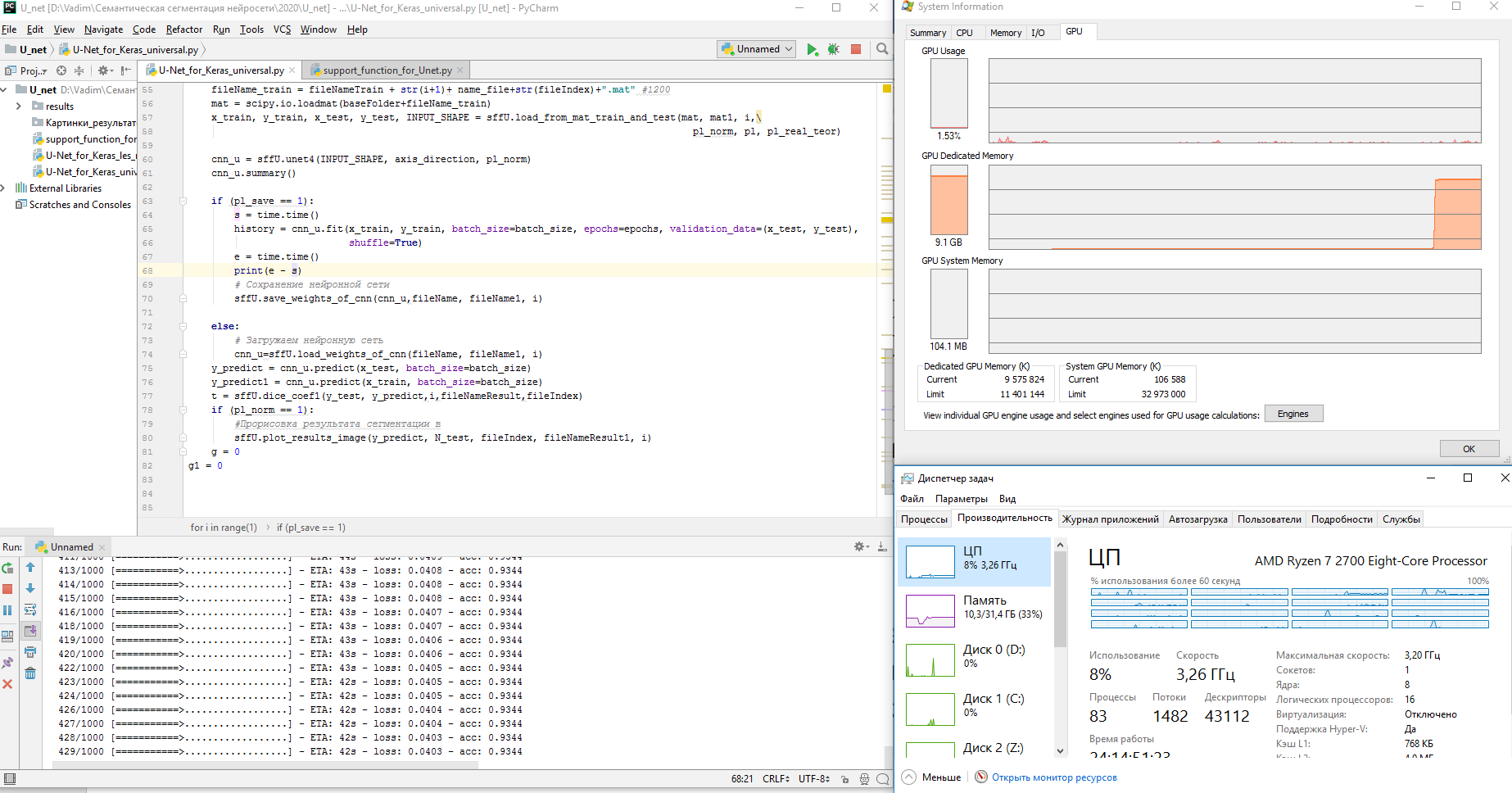


Рисунок 6 – Обучение U-net

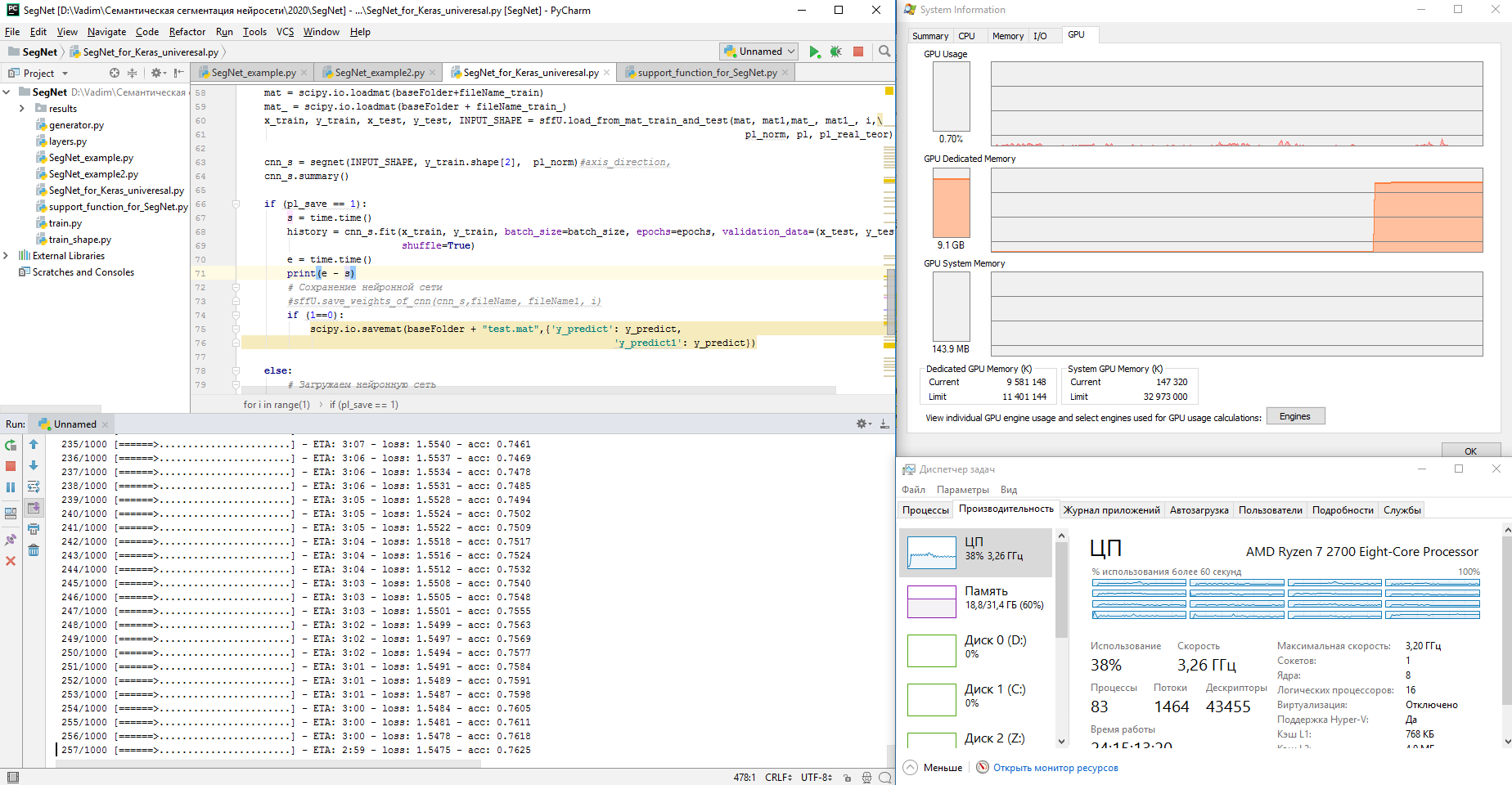


Рисунок 7 – Обучение SegNet

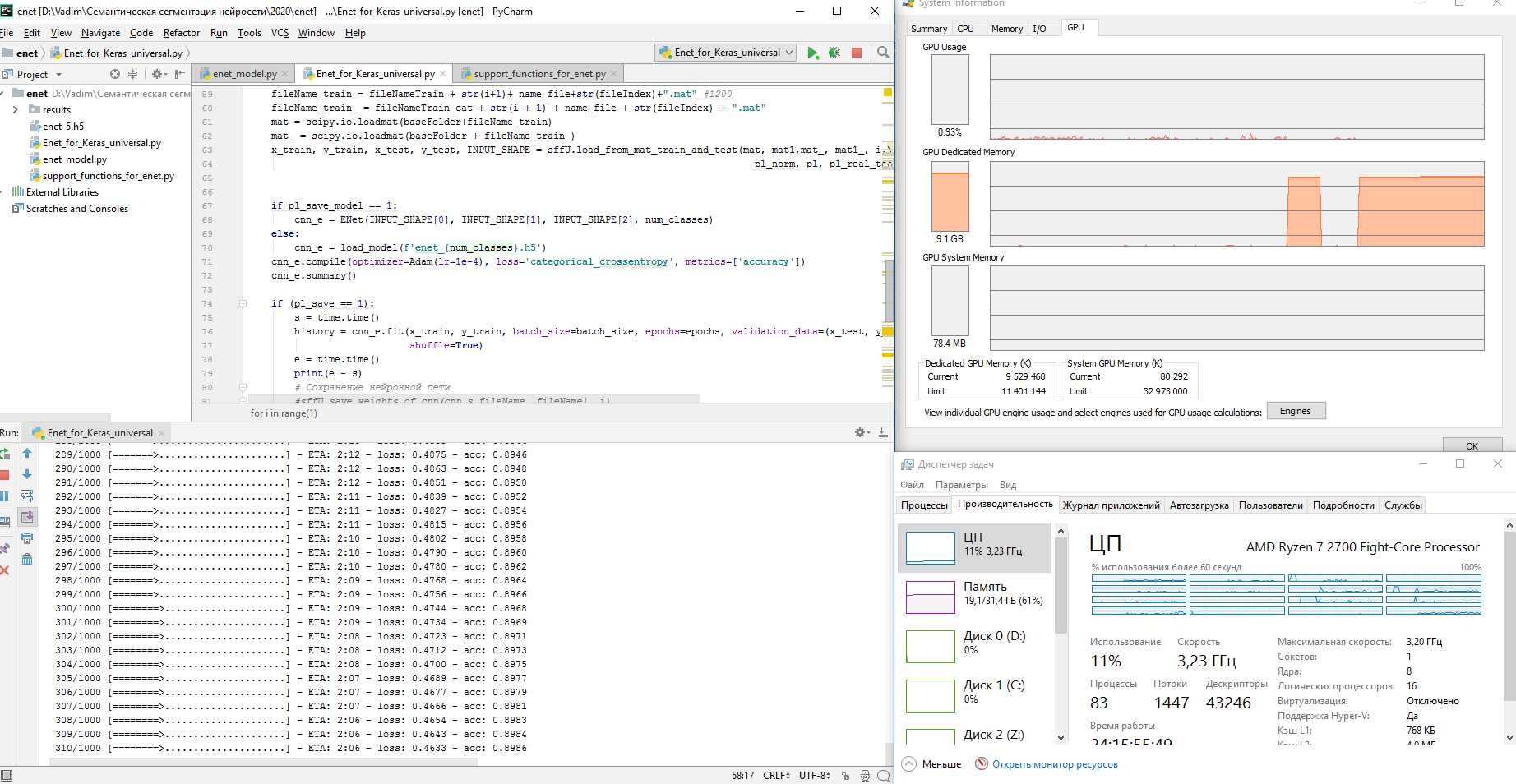


Рисунок 8 – Обучение Enet

В таблице 1 приведены результаты по точности обучения трех видов нейронных сетей при 10 эпохах.

Таблица 1 - Результаты по точности обучения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Архитектура нейронной сети | loss | accuracy | val\_loss | val\_accuracy | Время на одну проекцию, с |
| U-net | 0.0247 | 0.9349 | 0.0250 | 0.9340 | 614,64937 |
| SegNet | 0.5169 | 0.9449 | 0.5086 | 0.9440 | 2210,02376 |
| Enet | 0.0125 | 0.9952 | 0.0121 | 0.9953 | 1334,42567 |

На основании результатов точности работы (таблица 3) для распознавания была выбрана архитектура Enet. В то же время быстрее всего произошло обучение U-net, что связано с меньшей размерностью матриц-меток.

**ЗАКЛЮЧЕНИЕ**

В результате лабораторный работы выполнены и обобщены в виде таблицы результаты обучения трех архитектур нейронных сетей на GPU. При проверке точности обучения лучший результат показывает архитектура Enet.

**Список используемых источников**

1. D. Cohen-Steiner and J. Morvan, "Restricted delaunay triangulations and normal cycle", 19th Annu. ACM Sympos. Comput. Geom., 2003, pp. 312-321.

2. Ronneberger, O. U-net: Convolutional networks for biomedical image segmentation [Text] / O. Ronneberger, P. Fischer, T. Brox // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015. – 9351. – P. 234-241.

3. Badrinarayanan Vijay, Kendall Alex, Cipolla Roberto. Segnet: A deep convolutional encoder-decoder architecture for image segmentation // IEEE transactions on pattern analysis and machine intelligence. ––2017. –– Vol. 39, no. 12. –– P. 2481–2495.

4. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation / Adam Paszke, Abhishek Chaurasia, Sangpil Kim, Eugenio Culurciello // CoRR. –– 2016. –– Vol. abs/1606.02147. –– 1606.02147.