



УДК 519.764

К. Р. Ахметзянов, А. А. Южаков  
Пермский национальный исследовательский  
политехнический университет (ПНИПУ)

## Сравнение сверточных нейронных сетей для задач сортировки мусорных отходов

*Показан процесс выбора сверточной сети для микрокомпьютера RaspberryPi, который является частью фандомата – устройства для сбора и сортировки пластиковых бутылок и алюминиевых банок. Для обучения были выбраны три нейронные сети: AlexNet, SqueezeNet и MobileNet. Обучение производилось с помощью Transfer Learning двумя способами: заменой последнего слоя классификатора с последующим обучением и настройкой всей сети. В качестве фреймворка для запуска нейронных сетей был выбран Caffe как наиболее популярный. Приведены результаты правильного распознавания для проверочной выборки, состоящей из 2300 фотографий, включающих фотографии банок, бутылок и «прочего мусора», а также более подробно описан результат проверки на небольшой тестовой выборке, состоящей из 30 фотографий. В работе приводятся временные характеристики работы нейронных сетей на компьютере и на RaspberryPi. В ходе экспериментов показано, что самый высокий процент правильного распознавания имеет сеть MobileNet, минимальное время, необходимое для обработки одного изображения на RaspberryPi, имеет сеть SqueezeNet.*

### Классификации объектов, нейронные сети, сверточные нейронные сети, глубокое обучение, компьютерное зрение, сортировка отходов

Проблема утилизации отходов в настоящее время особенно актуальна. Для того чтобы утилизировать отходы, их необходимо перед переработкой отсортировать. Сортировать можно на этапе общего сбора мусора и после него.

Предварительную сортировку предлагается вести фандоматом – устройством для сбора пластиковых бутылок и алюминиевых банок, а также отбраковки (не сбора) «прочего мусора», при которой мусор будет сортироваться на этапе сбора. Фандомат состоит из микрокомпьютера (RaspberryPi), к которому подключена видеочамера, исполнительных механизмов сортировки и двух контейнеров, в которых будут находиться бутылки и банки. Сортировка заключается в том, чтобы по фотографии объекта, который был помещен в фандомат, определить, пластиковая ли бутылка этот объект, алюминиевая банка или ни то ни другое. Проблема сортировки заключается в том, что необходимо в течение секунды классифицировать объект на устройстве с ограниченным количеством ресурсов. Для этого предлагается использовать глубокую сверточную сеть (далее будет

описано, почему именно ее) с использованием фреймворка Caffe [1]. В работе приведено сравнение качества распознавания для трех сетей: AlexNet, SqueezeNet, MobileNet.

**Причины выбора сверточных нейронных сетей.** Сверточные сети – это разновидность нейронных сетей, которые состоят из фильтров (сверток). Каждый такой фильтр с помощью обучения нейронной сети учится различать информацию от различных линий и закорючек (первые слои) до определенных образов (последние слои) [2].

Первые сверточные нейронные сети были придуманы и применены Я. Лекуном в 1989 г. [3] для распознавания рукописных символов. Такая сеть называется LeNet. Однако тогда нейронные сети не получили широкого применения, потому что не было достаточного количества вычислительных мощностей для быстрой работы сетей.

Вновь к сверточным нейронным сетям вернулись в 2012 г., когда команда Алекса Крижевского заняла первое место в соревновании ILSVRC-

2012 [4], причем по правильности распознавания их решение значительно опередило лучшее решение 2011 г. (ILSVRC-2011). Сверточная сеть Алекса Крижевского называется AlexNet [5] и очень похожа на LeNet.

В дальнейшем на основе сети AlexNet была создана сеть Network-in-Network (NiN) [6], в которой стали применяться малые фильтры (свертки) размером  $1 \times 1$  и сама сеть состоит из последовательно соединенных нескольких сетей, что увеличило правильность распознавания. Дальнейшее развитие состоит в создании сети VGG [7], которая также структурирована из малых фильтров ( $3 \times 3$  и  $1 \times 1$ ) и еще большего числа слоев. В дальнейшем идею увеличения количества слоев развила команда Google, которая создала сеть GoogLeNet [8]. В этой сети также применяются малые фильтры  $3 \times 3$  и  $1 \times 1$  и на основе идеи сети NiN сеть GoogLeNet состоит из последовательно соединенных маленьких сетей (модулей), которые называются Inception.

Такие глубокие сети даже стали отдельным направлением в машинном обучении под названием «глубокое обучение» (Deep Learning) [9]. Эти сети показывают высокую степень правильного распознавания, но скорость их работы достаточно низкая. В связи с развитием мобильных технологий и вычислений на мобильных устройствах (смартфонах) разработчики нейронных сетей захотели перенести свои нейронные сети на смартфоны. Поэтому были созданы сети SqueezeNet [10] и MobileNet [11]. В SqueezeNet содержится большое количество слоев, состоящих из малых сверток, но при этом количество параметров в несколько раз меньше (следовательно, меньше и количество вычислений), чем у AlexNet, в процентах правильная классификация такая же, как у AlexNet. В MobileNet, как и в SqueezeNet, большое количество слоев с маленькими свертками, но количество параметров еще меньше, чем в SqueezeNet, и в смартфонах эта нейронная сеть работает быстро.

В статье исследуется правильность распознавания банок и бутылок с помощью сетей AlexNet, SqueezeNet и MobileNet, так как у этих сетей скорость работы на RaspberryPi удовлетворяет поставленной задаче (время обработки меньше 1 с).

**Процесс обучения нейронных сетей.** Обучение сетей производилось на графической карте Nvidia GeForce GT 740M, состоящей из 384 ядер с тактовой частотой 993 МГц и графической памятью 2048 Мбайт. Для обучения был выбран Caffe – популярный фреймворк с большим сообществом раз-

работчиков [12] и моделью – результатом обучения, которую легко перенести с одного компьютера на другой и можно запустить на CPU, даже если обучение производилось на GPU. Для запуска моделей на RaspberryPi используется модуль dnn из OpenCV. OpenCV скомпилирован с использованием NEON и VFPV3, что увеличивает скорость работы нейронной сети на RaspberryPi. Предварительно обученная сеть проверялась на компьютере с центральным процессором Intel Core i7-3610QM с 8 ядрами, базовой тактовой частотой 2.3 ГГц и оперативной памятью 8 Гбайт. Обученная сеть проверялась на RaspberryPi 3 с центральным процессором Broadcom BCM2837 с 4 ядрами, частотой 1.2 ГГц и 1 Гбайт оперативной памяти. Обучение производилось на 500 фотографиях бутылок и 500 фотографиях банок, которые были скачаны из поисковых запросов Google, а также в качестве изображений «прочего мусора» использовались 10 000 картинок из базы данных UKBench [13]. Бутылки и банки в обучающей выборке в основном расположены вертикально, горлышком вверх, и немятые. Проверка во время обучения осуществляется на проверочной выборке, состоящей из 150 фотографий бутылок, столько же – банок и 2000 фотографий «прочего мусора». Эта выборка состоит из фотографий, которых нет в обучающей выборке. Небольшая выборка для тестирования состоит из 10 фотографий бутылок, столько же банок и столько же «прочего мусора». Эта выборка содержит фотографии бутылок и банок горлышком вертикально вниз и вверх, а также мятых и немятых банок и бутылок.

Для обучения использовалась технология «передачи знания» (Transfer Learning) [14]. Суть ее состоит в том, что нейронные сети обучаются не «с нуля», т. е. не со случайных значений весов, а с уже обученной модели. Модели могут быть обучены как на наборе фотографий ImageNet [15], так и на других наборах фотографий. Без использования технологии Transfer Learning обучать нейронные сети пришлось бы на гораздо большем наборе данных и такое обучение длилось бы несколько недель. Обученные модели AlexNet, SqueezeNet получены из Model Zoo [16]; MobileNet – из [17].

Transfer Learning подразделяется на два вида [14]:

1. Применение сверточной сети для извлечения признаков. Все веса обученной сети переносятся в новую сеть и остаются в том виде, в котором они есть, но последний слой (классификатор) отбрасывается, и создается новый класси-

фикатор, который обучается «с нуля» на новом наборе данных.

2. Точная настройка (fine tuning) сверточной сети. Все веса обученной сети переносятся в новую сеть и корректируются с учетом нового набора данных, т. е. обучение производится не со случайного состояния, а с сети, которая уже умеет отличать объекты какого-либо набора данных.

Далее будут рассмотрены оба варианта обучения для всех трех сверточных сетей.

**Результаты эксперимента.** В Caffe процесс обучения определяется количеством итераций. Количество итераций – это количество частей обучающей выборки (batch), на которых произошло обучение [18]. Для обучения разных сетей применялись части разного размера из-за ограничений памяти видеокарты (чем больше этот размер, тем больше требуется памяти). На наш взгляд более показательным является количество эпох, которое показывает, сколько полных обучающих выборок прошло через процесс обучения. Размер части выборки влияет на время, которое уходит у нейронной сети на обработку выборки (чем больше этот размер, тем больше время), на разброс правильного распознавания (чем меньше этот размер (в процентах), тем больше разброс, чем больше этот размер, тем более плавно увеличивается правильное распознавание (в процентах)), а также нейронная сеть при слишком маленьком размере части выборки может не обучиться, так как произойдет не постепенное увеличение правильного распознавания, а сильный разброс от итерации к итерации [19].

В табл. 1 указаны размеры частей, которые использовались для обучения по каждой сети. Столбцом «Один слой» отмечена сеть, у которой обучен только классификатор. Столбцом «Вся сеть» отмечена сеть, у которой обучены все слои.

Таблица 1

Сеть	Размер частей	
	Один слой	Вся сеть
AlexNet	64	64
SqueezeNet	32	32
MobileNet	8	8

Для проверки сети брались модели, которые достигали максимума правильности распознавания для проверочной выборки. Этот максимум достигается для разных нейронных сетей по-разному, но (исходя из опыта) после 1000-й итерации правильное распознавание практически не меняется, поэтому количество итераций обучения для всех моделей определяется 1000. На рис. 1 представлен график изменения правильности распознавания для проверочной выборки нейронной сети AlexNet при точной настройке всей сети.

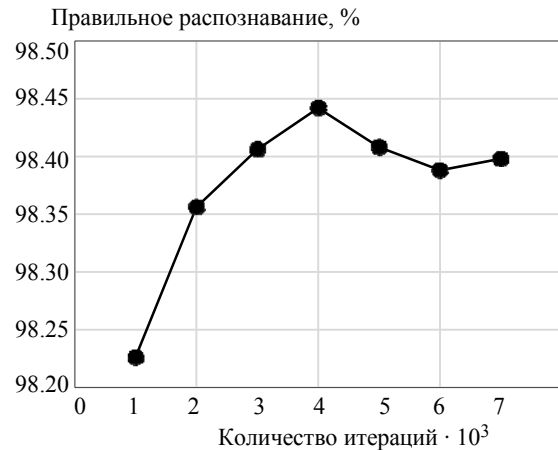


Рис. 1

Как видно из рис. 1, правильное распознавание в сети AlexNet достигло примерно 98.2 % за 1000 итераций. Дальнейшие итерации меняют правильное распознавание в пределах 0.1 %.

На рис. 2 изображен график изменения правильности распознавания для проверочной выборки нейронной сети SqueezeNet при точной настройке всей сети.

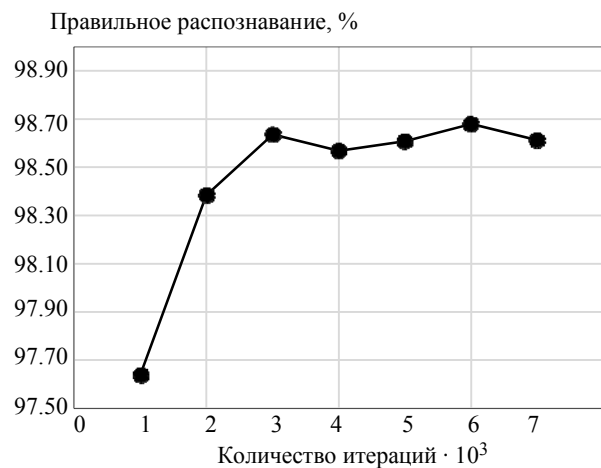


Рис. 2

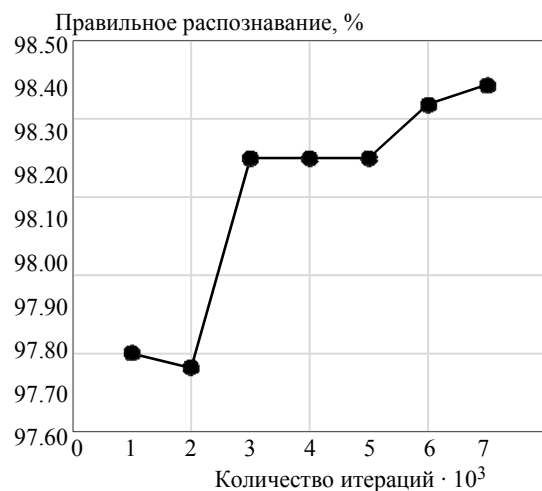


Рис. 3

Таблица 2

Сеть	Итерация	
	Один слой	Вся сеть
AlexNet	4000	4000
SqueezeNet	1000	6000
MobileNet	2000	7000

Таблица 3

Сеть	Правильное распознавание, %	
	Один слой	Вся сеть
AlexNet	97.84	98.44
SqueezeNet	98.11	98.64
MobileNet	97.85	98.39

Как видно из рис. 2, правильное распознавание в сети SqueezeNet достигло примерно 97.6 % за 1000 итераций, далее 2000-я итерация увели-

чила правильное распознавание примерно на 0.8 %, 3000-я итерация – примерно на 0.2 %. Дальнейшие итерации меняют правильное распознавание в пределах 0.1 %.

На рис. 3 изображен график изменения правильности распознавания для проверочной выборки нейронной сети MobileNet при точной настройке всей сети.

Как видно из рис. 3, правильное распознавание в сети MobileNet достигло примерно 97.7 % за 1000 итераций, далее 2000-я итерация уменьшила правильное распознавание примерно на 0.05 %, 3000-я – увеличила примерно на 0.5 %; 4000-я и 5000-я итерации почти не меняют пра-

Таблица 4

Параметр		AlexNet				SqueezeNet				MobileNet			
		Один слой		Вся сеть		Один слой		Вся сеть		Один слой		Вся сеть	
		Прав.	%	Прав.	%	Прав.	%	Прав.	%	Прав.	%	Прав.	%
Бутылки	Образец 1	1	100	1	100	1	100	1	100	1	100	1	100
	Образец 2	1	47	1	59	1	77	1	96	0	47	1	49
	Образец 3	1	100	1	100	1	100	1	100	1	98	1	100
	Образец 4	1	99	1	100	1	77	1	90	0	70	1	83
	Образец 5	0	57	0	100	0	49	0	81	0	45	0	88
	Образец 6	1	100	1	100	1	100	1	100	1	96	1	100
	Образец 7	0	100	0	100	0	100	0	99	0	40	0	78
	Образец 8	1	98	1	100	0	74	1	100	1	95	1	100
	Образец 9	1	100	1	100	1	99	1	100	1	100	1	100
	Образец 10	1	63	1	75	1	99	0	93	0	65	1	85
Банки	Образец 11	1	100	1	100	1	96	1	100	1	97	1	100
	Образец 12	1	99	1	100	1	90	1	100	1	100	1	100
	Образец 13	1	97	1	98	0	53	1	69	1	84	1	80
	Образец 14	1	75	1	97	1	98	1	100	1	95	1	100
	Образец 15	1	100	1	100	1	87	1	100	1	98	1	72
	Образец 16	1	98	1	100	1	80	1	100	1	100	1	100
	Образец 17	1	95	1	100	1	99	1	100	1	98	1	100
	Образец 18	1	99	1	100	1	100	1	100	1	100	1	100
	Образец 19	1	97	1	100	1	91	1	100	1	98	1	100
	Образец 20	0	99	0	99	0	100	0	98	1	100	1	87
Мусор	Образец 21	1	99	1	100	1	100	1	100	1	100	1	100
	Образец 22	1	100	1	96	1	99	1	99	1	93	1	98
	Образец 23	1	100	1	100	1	100	1	100	1	99	1	100
	Образец 24	1	100	1	100	1	100	1	100	1	100	1	100
	Образец 25	1	99	1	100	1	100	1	100	1	95	1	98
	Образец 26	1	99	1	100	1	100	1	100	1	99	1	100
	Образец 27	1	100	1	100	1	100	1	100	1	99	1	100
	Образец 28	1	100	1	100	1	100	1	100	1	93	1	100
	Образец 29	1	100	1	100	1	100	1	100	1	100	1	100
	Образец 30	1	100	1	100	1	100	1	100	1	100	1	100
Правильность распознавания, %		90		90		83		87		83		93	

вильное распознавание, 6000-я увеличила правильное распознавание примерно на 0.15 %. Дальнейшие итерации меняют правильное распознавание в пределах 0.1 %.

В табл. 2 показано количество итераций, на которых достигается максимум правильного распознавания по каждой сети.

В табл. 3 показано максимальное правильное распознавание по каждой сети для проверочной выборки.

Как видно из табл. 3, максимальное правильное распознавание для проверочной выборки достигается в сети SqueezeNet со всеми обученными слоями и равно 98.64 %.

В табл. 4 показаны результаты распознавания по каждой сети для небольшой выборки тестирования. В столбце «Прав.» отмечено, правильно ли нейронная сеть распознавала образец (1 – правильно, 0 – неправильно). В столбце «%» показана степень уверенности нейронной сети в том, к какому классу она отнесла образец.

Как видно из табл. 4, правильное распознавание для небольшой тестовой выборки наиболее высоко у сети MobileNet со всеми обученными слоями.

В табл. 5 описаны характеристики образцов. Если образец обладает свойством, указанным в названии столбца, то в ячейке ставится «+», если не обладает, то «-».

Таблица 5

Образец	Горлышком вверх	Мягкий
1	+	-
2	-	-
3	+	+
4	-	+
5	-	-
6	+	+
7	-	-
8	+	-
9	+	-
10	-	-
11	+	-
12	-	-
13	+	+
14	+	+
15	+	+
16	+	+
17	+	-
18	-	-
19	+	+
20	+	-

Как видно из табл. 4 и 5, ни одна сеть не смогла распознавать образцы 5 и 7 – перевернутую немятую пластиковую бутылку.

В табл. 6 показано время распознавания одного изображения по каждой сети на компьютере с CPU, характеристики которого указаны ранее, и на RaspberryPi.

Таблица 6

Сеть	Один слой	Вся сеть
Время/изображение на CPU, мс		
AlexNet	174	181
SqueezeNet	373	400
MobileNet	118	117
Время/изображение на RaspberryPi, мс		
AlexNet	887	900
SqueezeNet	323	317
MobileNet	748	725

Как видно из табл. 6 время распознавания у AlexNet и у MobileNet на RaspberryPi увеличилось в несколько раз, а у SqueezeNet – уменьшилось. Скорее всего, это вызвано хорошей оптимизацией OpenCV под процессоры ARM и меньшим количеством выходов сверток SqueezeNet (максимальное количество выходов у SqueezeNet равно 256, у MobileNet – 1024, у AlexNet – 384, также у AlexNet есть два полносвязных слоя с количеством выходов 4096, которые также могут увеличивать время обработки картинки).

Таким образом, правильное распознавание среди указанных в табл. 4 выше всего в сети MobileNet при настройке всех слоев. Минимальное время обработки одного изображения на RaspberryPi среди времен, указанных в табл. 6, имеет сеть SqueezeNet.

В исследовании представлен процесс выбора сверточной нейронной сети для микрокомпьютера RaspberryPi, который является частью фандомата. Сравнивались нейронные сети AlexNet, SqueezeNet и MobileNet с двумя вариантами обучения: обучение только классификатора и настройка всей сети. При сравнении правильной классификации всех сетей для небольшой тестовой выборки лучшее распознавание имеет сеть MobileNet при настройке всех слоев. Все сети, рассмотренные в этой статье, были запущены на RaspberryPi. В результате у всех сетей, кроме SqueezeNet, время работы увеличилось в несколько раз. Дальнейшие исследования могут идти в следующих направлениях:

1. Улучшение архитектуры нейронной сети MobileNet с целью увеличения правильного распознавания (в процентах).

2. Создание нейрона, способного достичь большего правильного распознавания (в процентах) в модифицированной нейронной сети.

3. Увеличение обучающей выборки.
4. Добавление предварительной обработки изображения.

5. Поиск более экономичных по затратам времени на обработку изображений способов работы нейронной сетей для RaspberryPi.

## СПИСОК ЛИТЕРАТУРЫ

1. Caffe: Convolutional Architecture for Fast Feature Embedding / Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell. URL: <https://arxiv.org/pdf/1408.5093.pdf> (дата обращения: 06.04.2018).
2. A Beginner's Guide To Understanding Convolutional Neural Networks. Pt. 2. URL: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/> (дата обращения: 21.01.2018).
3. Backpropagation applied to handwritten zip code recognition / Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel // Neural Computation. 1989. № 1 (4). P. 541–551.
4. Large Scale Visual Recognition Challenge 2012. URL: [www.image-net.org/challenges/LSVRC/2012/results.html](http://www.image-net.org/challenges/LSVRC/2012/results.html) (дата обращения: 21.01.2018).
5. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // Proc. of the 25<sup>th</sup> Intern. Conf. on Neural Information Proc. Systems (NIPS'12) / Curran Associates Inc. 2012. Vol. 1. P. 1106–1114.
6. Lin M., Chen Q., Yan S. Network in network // Proc. of the 2<sup>nd</sup> Intern. Conf. on Learning Representations 2014. Banff. Canada, URL: <https://arxiv.org/pdf/1312.4400.pdf> (дата обращения: 06.04.2018).
7. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. URL: <https://arxiv.org/pdf/1409.1556.pdf> (дата обращения: 06.04.2018).
8. Going deeper with convolutions / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich // Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Boston, USA, 2015. P. 1–9.
9. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 с.
10. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. arXiv препринт / F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer. arXiv:1602.07360, 2016. URL: <https://arxiv.org/pdf/1602.07360.pdf> (дата обращения: 06.04.2018).
11. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam. URL: <https://arxiv.org/pdf/1704.04861.pdf> (дата обращения: 06.04.2018).
12. DL4J, Torch7, Theano and Caffe. URL: <https://deeplearning4j.org/compare-dl4j-tensorflow-pytorch> (дата обращения: 21.01.2018).
13. UKBench Dataset. URL: <https://archive.org/details/ukbench> (дата обращения: 30.12.2017).
14. Transfer Learning. URL: <http://cs231n.github.io/transfer-learning/> (дата обращения: 21.01.2018).
15. Imagenet: A large-scale hierarchical image database / J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei // Proc. CVPR, 2009.
16. Model Zoo. URL: <https://github.com/BVLC/caffe/wiki/Model-Zoo> (дата обращения: 29.01.2018).
17. MobileNet-Caffe. URL: <https://github.com/shicai/MobileNet-Caffe> (дата обращения: 29.01.2018).
18. Epoch vs Batch Size vs Iterations. URL: <https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9> (дата обращения: 21.01.2018).
19. Neural Network 3. URL: <http://cs231n.github.io/neural-networks-3/> (дата обращения: 21.01.2018).

---

K. R. Akhmetzyanov, A. A. Yuzhakov  
Perm National Research Polytechnic University Saint

## CONVOLUTIONAL NEURAL NETWORKS COMPARISON FOR WASTE SORTING TASKS

*We show convolutional neural network choosing process for RaspberryPi microcomputer. These networks are run on RaspberryPi, which is reverse vending machine part – device for collect and sorting of aluminum cans and plastic bottles. We chose and train three neural networks such as AlexNet, SqueezeNet, and MobileNet. For network training purpose, we use transfer learning, that is divided into two approaches: classifier replacement and retraining of convolutional neural network and weights fine-tuning of the pretrained network. As for framework, we use Caffe, because it is most popular. We evaluate these neural networks on validation set, which consists of 2300 photos. The validation set includes bottles, cans, and «other» trashes photos. We present the results of this evaluating. Also we evaluate networks on test set, which consists of 30 photos, and we present the results of these evaluating in detail. In this paper, the neural networks operation time on a computer and on RaspberryPi are given. Experiments are shown that MobileNet network has the highest accuracy and SqueezeNet has the minimum operation time per image on RaspberryPi.*

**Object classification, Caffe, Transfer Learning, neural networks, convolutional neural networks, Deep Learning, computer vision, waste sorting, RaspberryPi**

---