

# EyePoint P10

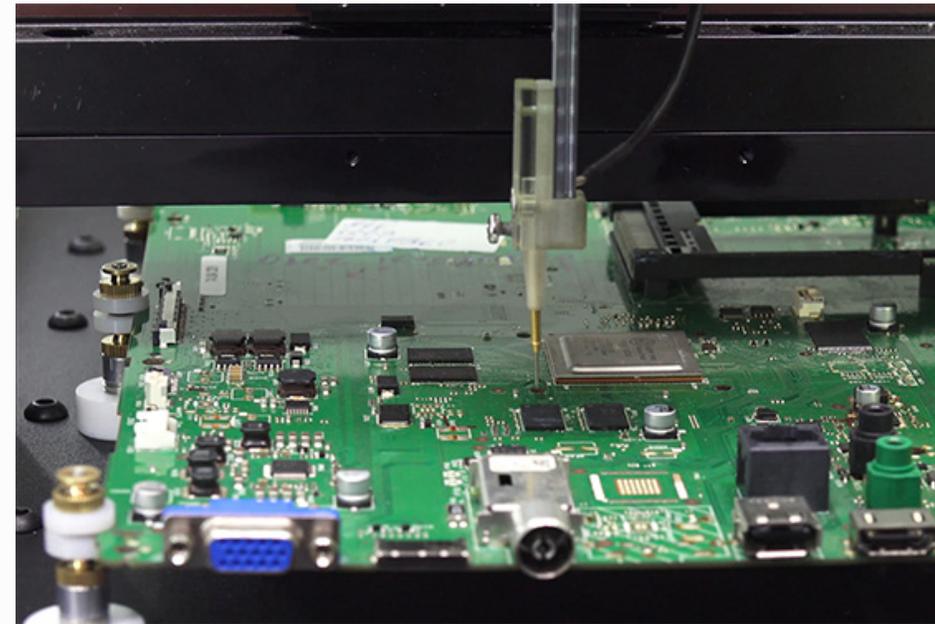
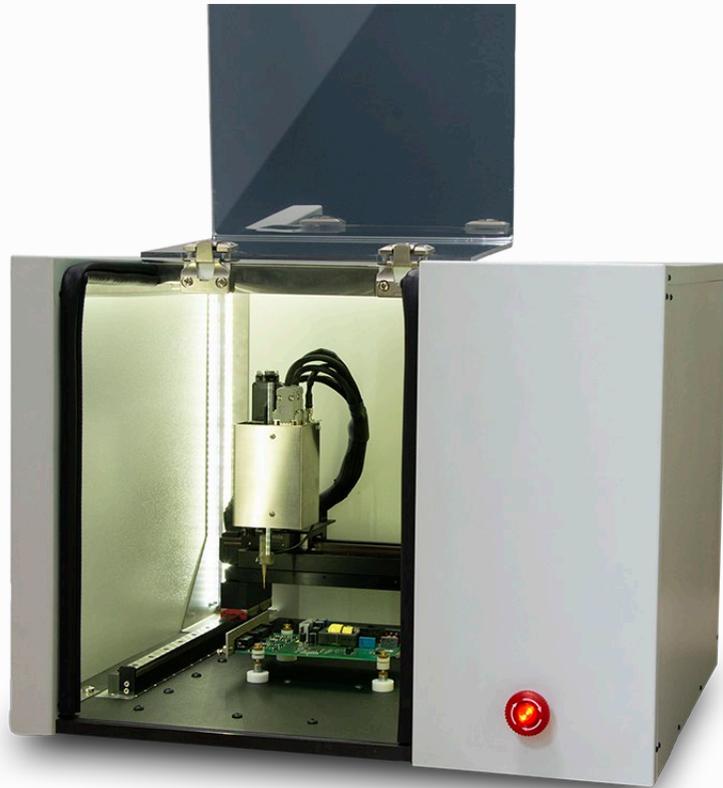
New look at troubleshooting  
with neural networks

Маракулин Андрей

# Описание установки



EyePointP10 - Система сигнатурного анализа для поиска неисправных электронных компонентов на печатных платах. Поиск производится в три этапа: сканирование платы, распознавание компонентов и тестирование компонентов.



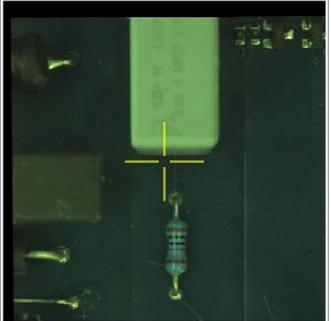
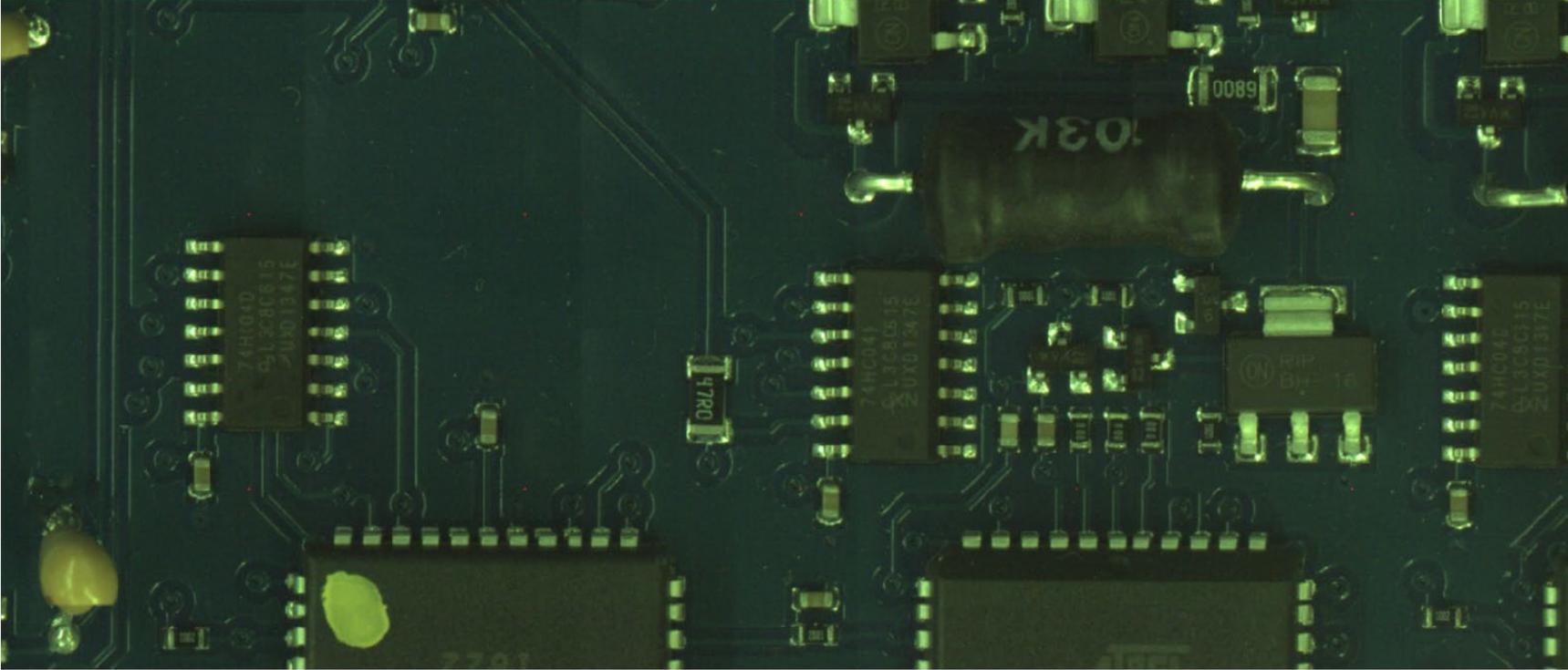
# Главное окно программы



EyePoint P10

File Tools

Calibration Scanning Testing



Scan Scan progress [0%]  
Find rotation [0%]  
Find elements Matching Skip  
Feature classifier [0%]  
Select PCB Matching [0%]  
Stop  
Manual tool  
Add Cancel  
PCB type: reference

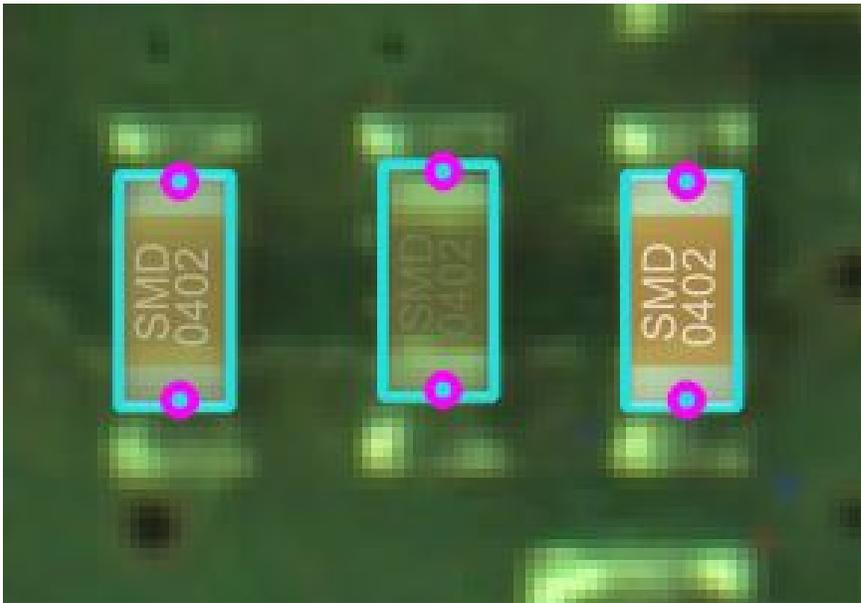
```
2020.05.02 14:42:13 DEBUG: Axis X borders: 0.0 115.30433654785156, position: 0.0
2020.05.02 14:42:13 DEBUG: Axis Y borders: 0.0 55.65214920043945, position: 0.0
2020.05.02 14:42:13 DEBUG: Axis X borders: 0.0 115.30433654785156, position: 0.0
2020.05.02 14:42:13 DEBUG: Axis Y borders: 0.0 55.65214920043945, position: 0.0
2020.05.02 14:42:13 DEBUG: Axis X borders: 0.0 115.30433654785156, position: 0.0
2020.05.02 14:42:13 DEBUG: Axis Y borders: 0.0 55.65214920043945, position: 0.0
2020.05.02 14:42:13 DEBUG: Axis X borders: 0.0 115.30433654785156, position: 0.0
2020.05.02 14:42:13 DEBUG: Axis Y borders: 0.0 55.65214920043945, position: 0.0
2020.05.02 14:42:14 DEBUG: Axis X borders: 0.0 115.30433654785156, position: 0.0
2020.05.02 14:42:14 DEBUG: Axis Y borders: 0.0 55.65214920043945, position: 0.0
2020.05.02 14:42:14 DEBUG: Axis X borders: 0.0 115.30433654785156, position: 0.0
2020.05.02 14:42:14 DEBUG: Axis Y borders: 0.0 55.65214920043945, position: 0.0
```

# Постановка задачи курсовой работы

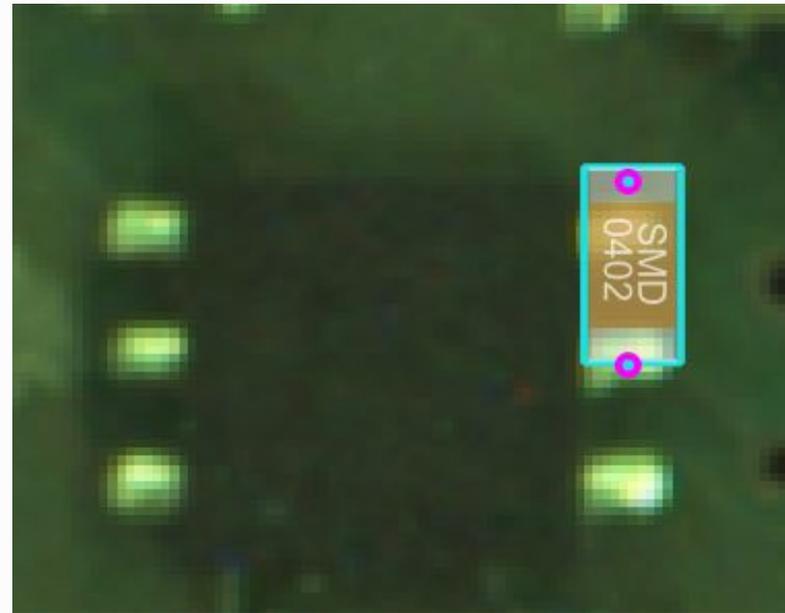


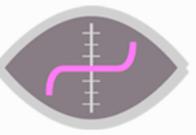
В курсовой работе ставится задача заменить существующий в программе метод распознавания компонентов на метод, использующий свёрточные нейронные сети и улучшить точность распознавания.

Пример корректного распознавания компонентов:



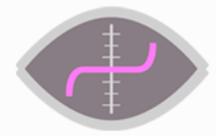
Пример ошибочного распознавания компонентов:



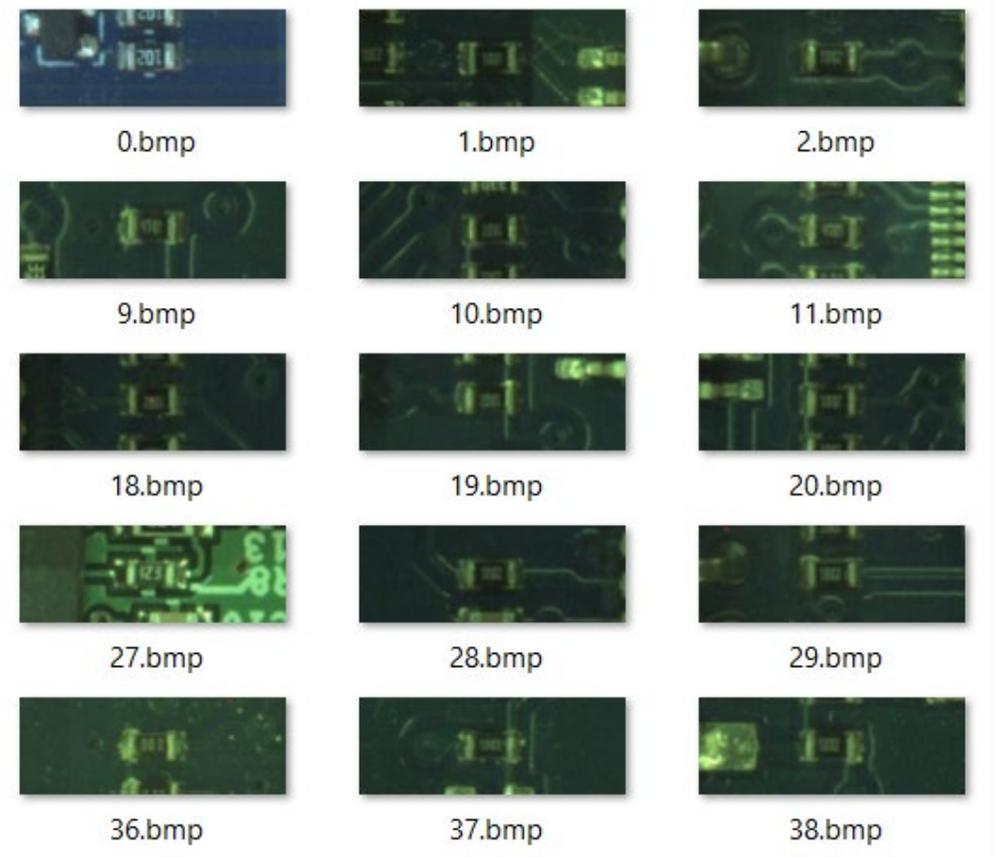
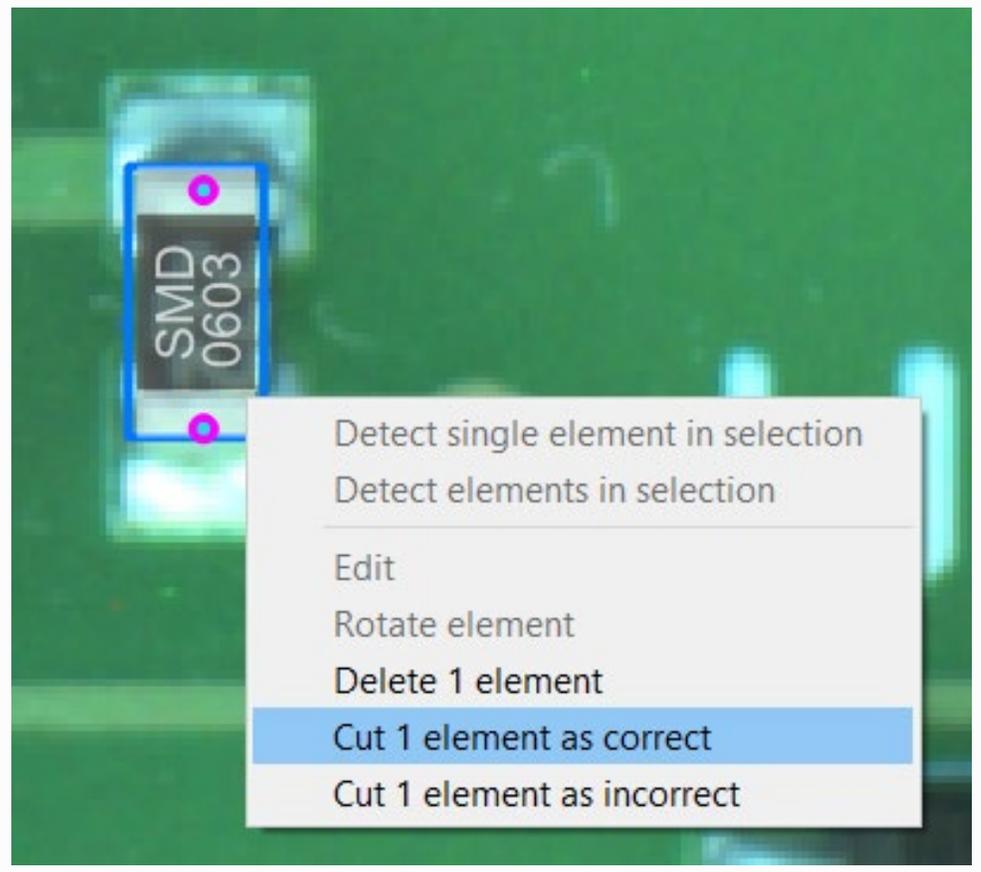


- Тестирование существующего функционала программы
- Сбор базы изображений компонентов, пригодной для обучения нейронных сетей
- Обучение нейронных сетей на собранной базе
- Интеграция нейронных сетей в функционал программы
- Сравнение результатов

# Сбор новой базы изображений



С помощью вспомогательной функции было собрано почти 25 000 изображений

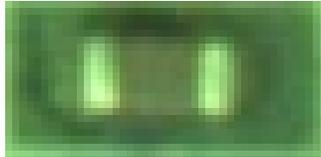


# Выбор компонентов



После сбора базы было выбрано 6 типов компонентов для обучения сетей, так как их набралось больше чем других

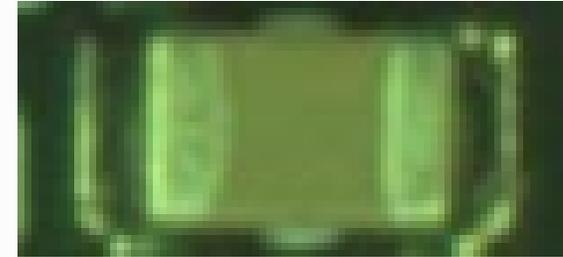
SMD0402\_C



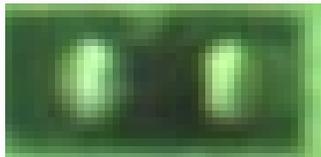
SMD0603\_C



SMD0805\_C



SMD0402\_R



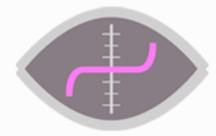
SMD0603\_R



SMD0805\_R

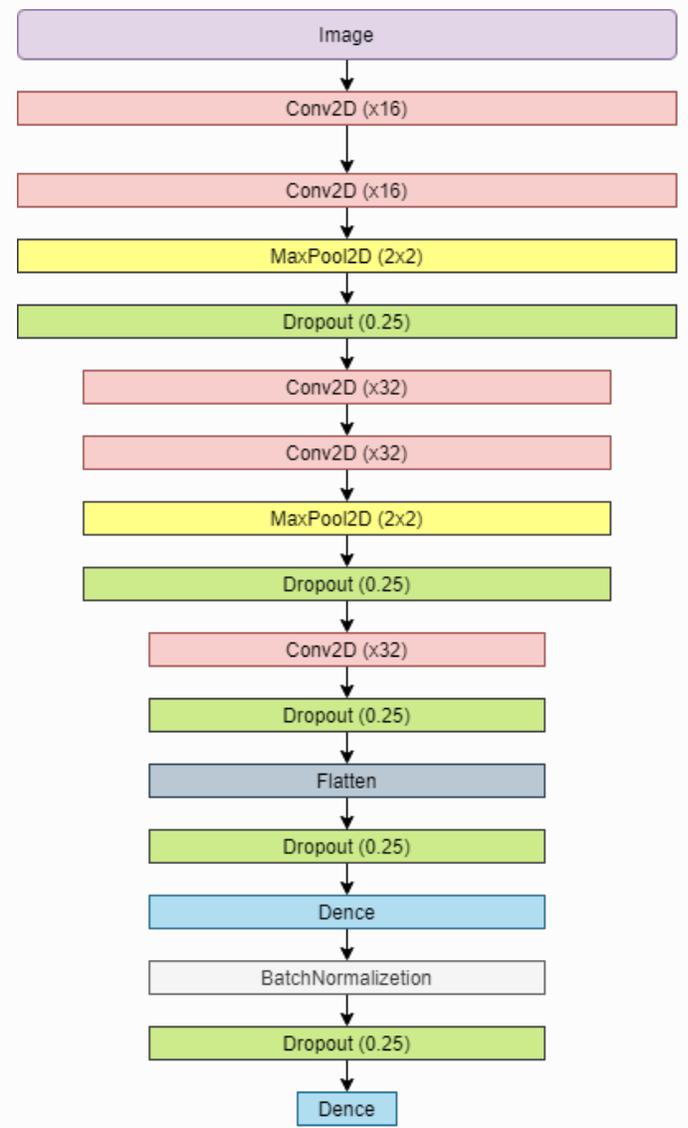
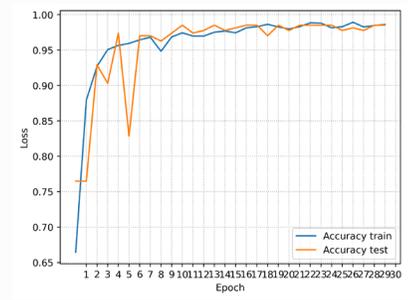
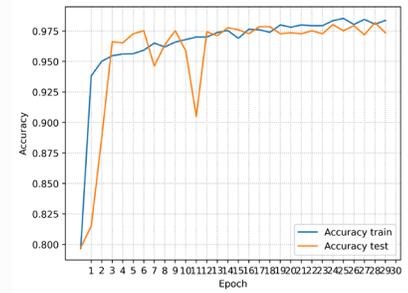


# Обучение сетей



Было решено разбить типы элементов на кластеры по разрешению, и обучать отдельные сети для каждого кластера

- SMD0402  
22x45 пикс.
- SMD0603  
24x66 пикс.
- SMD0805  
36x78 пикс.



# Интеграция сетей в программу



```
86 # ==== Run some code by start_mode
87 if start_mode == 'cut_normal':
88     candidates, candidates_img = cut_rotate_and_normalize(det, rgb_image, specified_hyp)
89     logging.debug(f"Found candidates: {len(candidates)}")
90 else:
91     logging.debug("Start mode unrecognized. Model skipped.")
92     return []
93
94 # ==== Do magic
95 logging.debug("Run predict...")
96 predict_arr = model.predict(candidates_img, use_multiprocessing=True)
97 del model
98 logging.debug("Done predict.")
99
100 # ==== Run some code by end_mode
101 if end_mode == 'end_normal':
102     threshold = det.trh_prob
103     result = end_normal(det, det.trh_prob, predict_arr, candidates, classes)
104 elif end_mode == 'end_thd':
105     threshold = define_threshold(det.trh_prob, data)
106     result = end_normal(det, threshold, predict_arr, candidates, classes)
107 else:
108     logging.debug("End mode unrecognized. Model skipped.")
109     return []
110
111 return result
```

# Интеграция сетей в программу



Название модели должно иметь такой формат:

`cut_normal.end_normal.21_22.h5`

Название Название Классы Расширение  
подготовительного завершающего для  
кода кода распознавания

- Из гипотез отбираются гипотезы с id классов, соответствующие этой модели и дополняются id классами-клонами, соответствующих поворотов.

Для унификации было предложено ассоциировать порядок классов в названии с предсказаниями нейронной сети.

В данном примере:

0 - предсказание это НЕ ЭЛЕМЕНТ, так как этот класс должен быть у всех сетей

1 - предсказание это 21 класс (SMD0402\_CL)

2 - предсказание это 22 класс (SMD0402\_R)

- Код соответствующий названию подготовительного кода должен подготовить гипотезы к подаче в predict нейронной сети
- Код соответствующий названию завершающего кода должен обработать результаты нейронной сети

Идентификация сетей по названию позволяет использовать совершенно различные сети: с разным разрешением, количеством выходов, настройками.

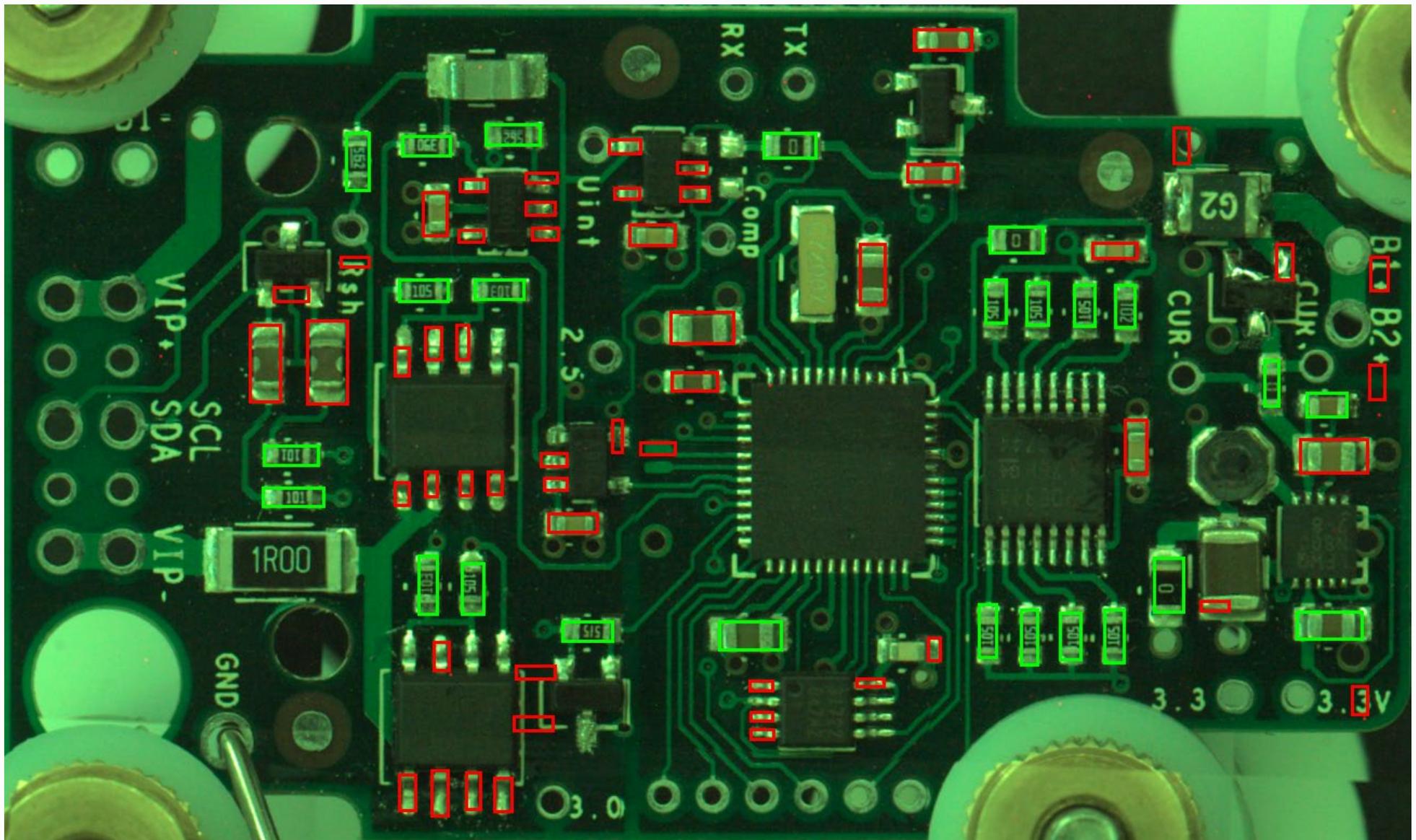
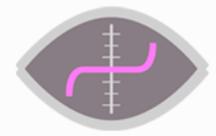
А так же менять их в процессе работы программы без её остановки!

 `cut_normal.end_normal.21_22.h5`

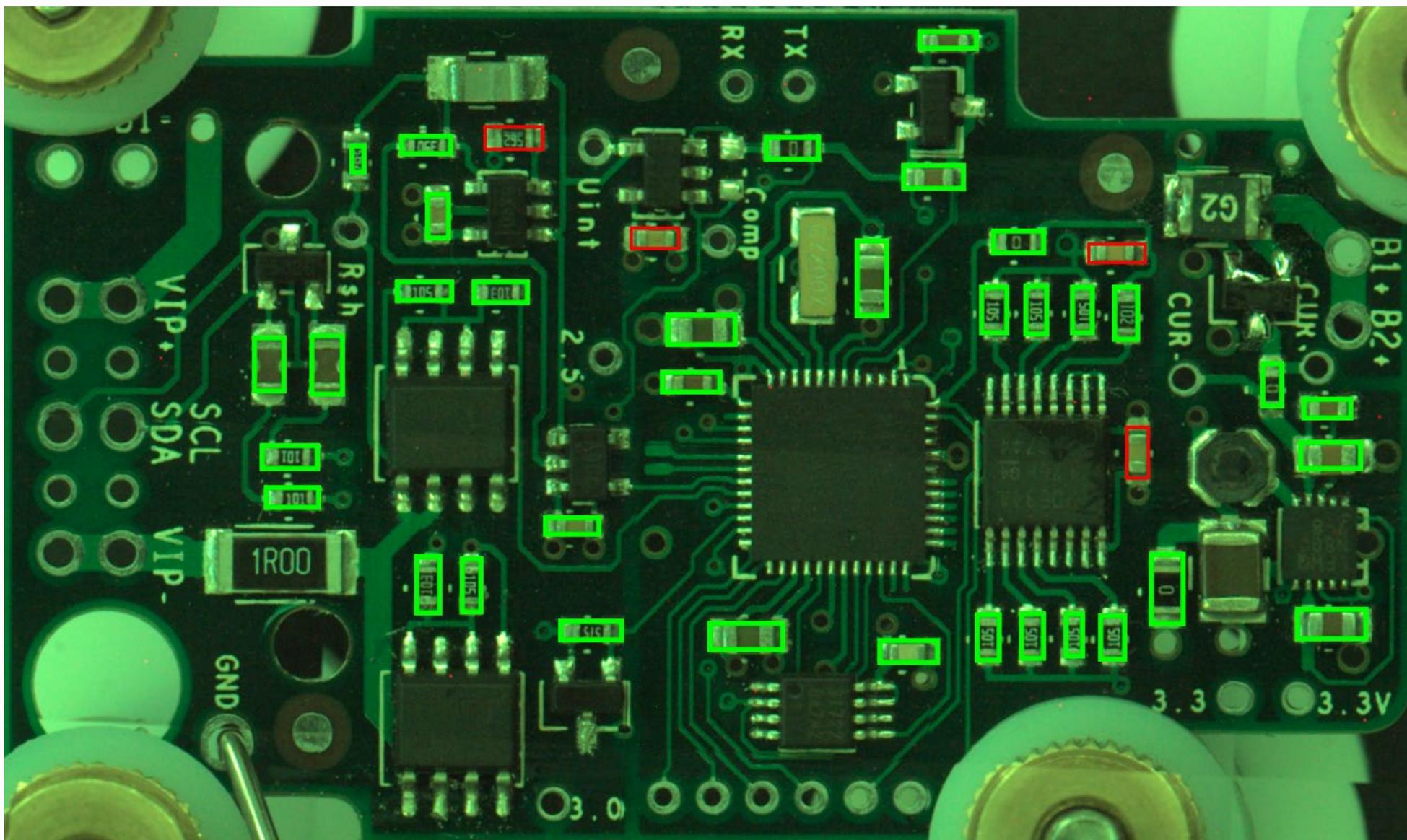
 `cut_normal.end_thd.0.95.23_24.h5`

 `cut_normal.end_thd.0.995.27_29.h5`

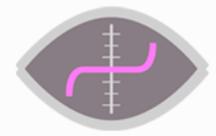
# Результат: прошлый метод



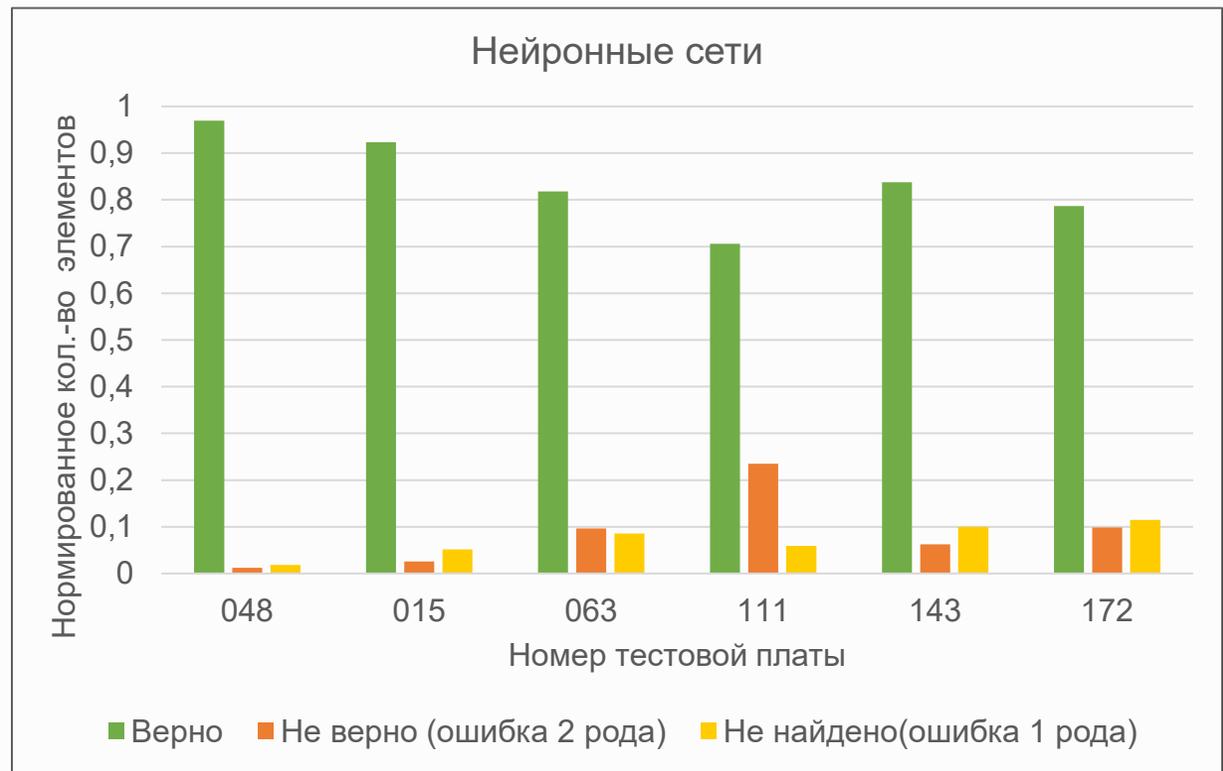
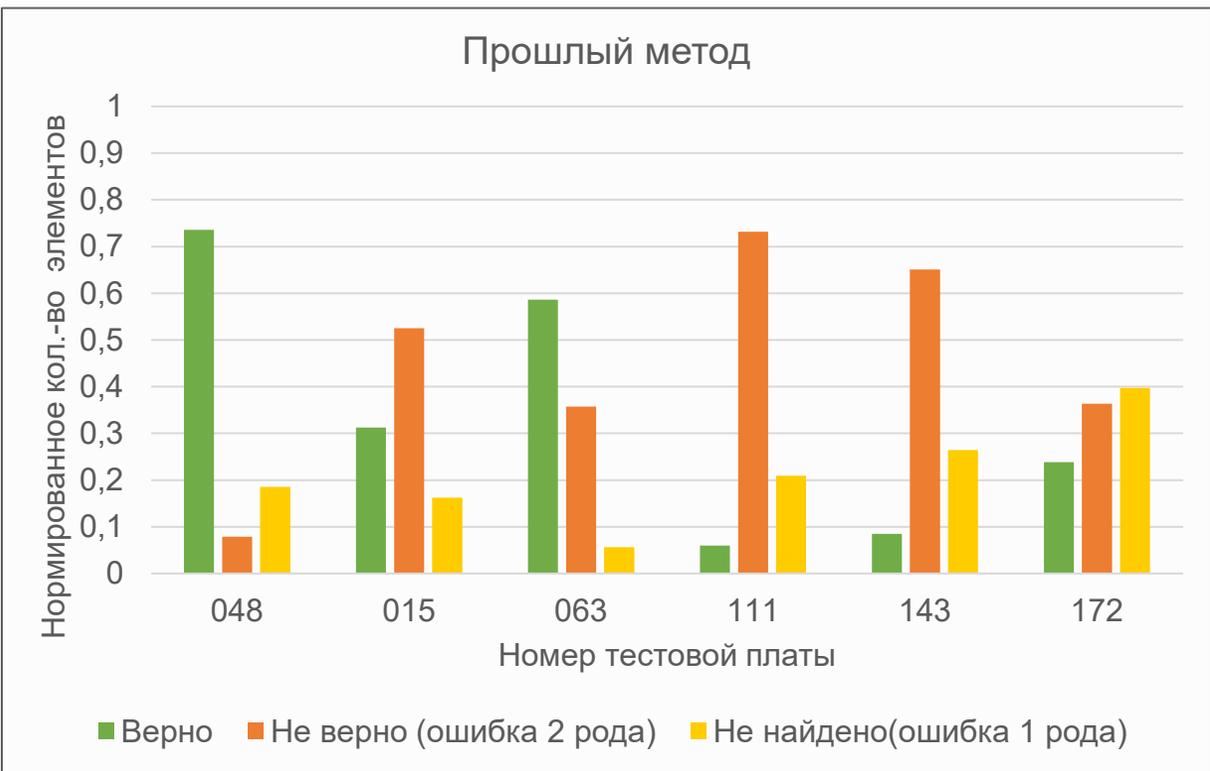
# Результат: нейронные сети



# Сравнение результатов



Чтобы объективно сравнить результаты можно посчитать количество правильных, неправильных и не найденных элементов на выборке из случайных плат



# Проверка гипотезы

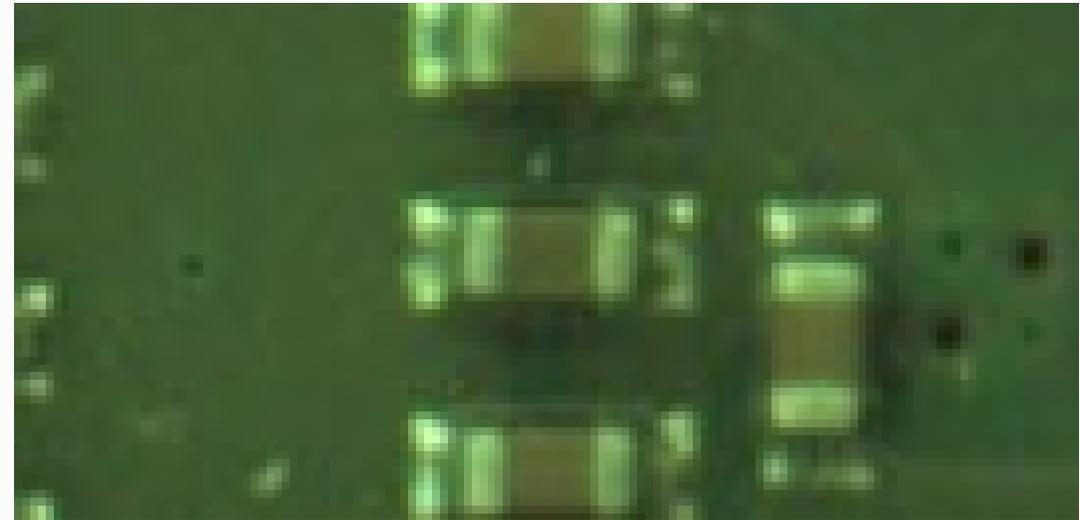


В процессе решения задачи была выдвинута гипотеза что увеличенная зона (увеличенный контекст) изображения вокруг элемента поможет улучшить точность нейронной сети.

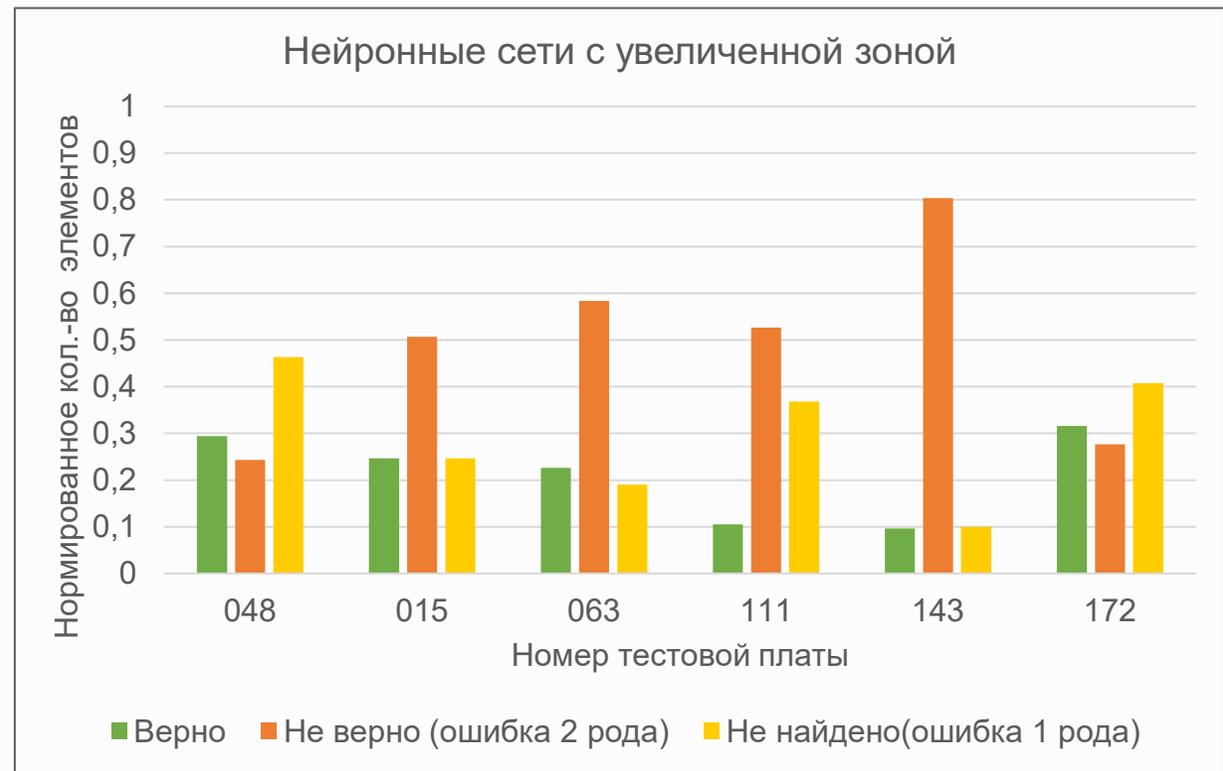
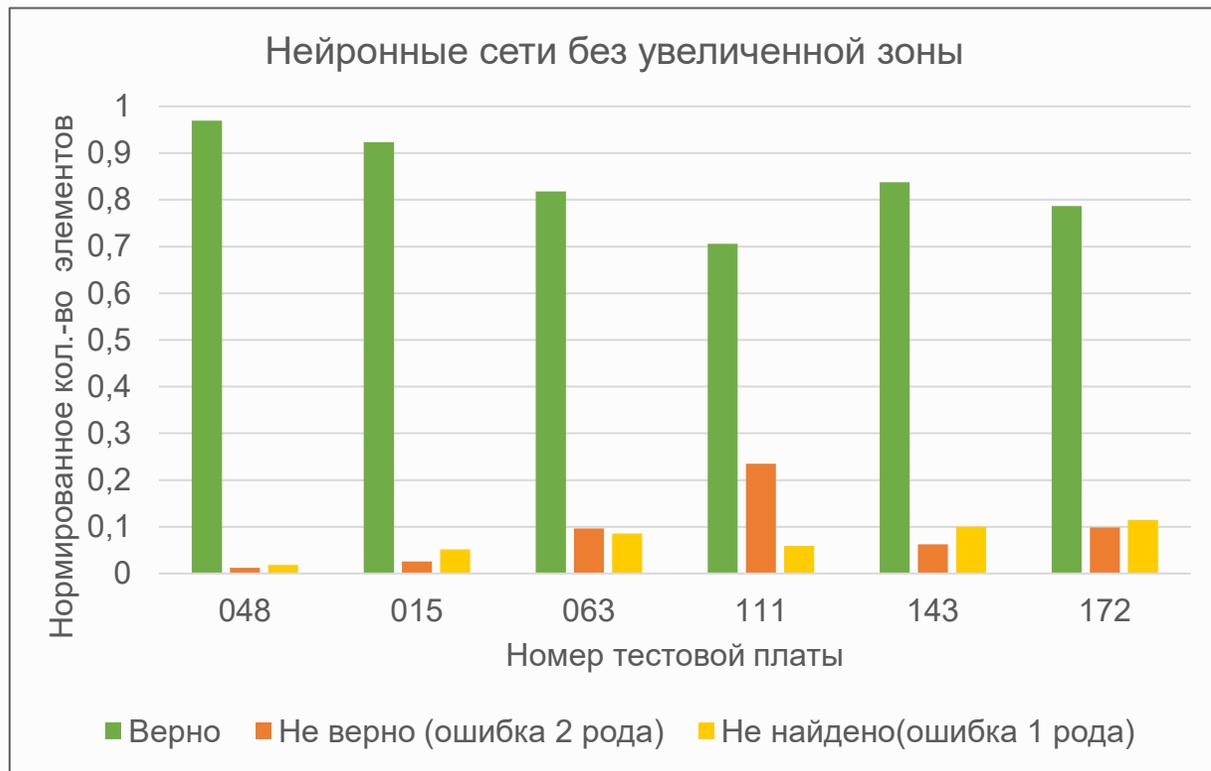
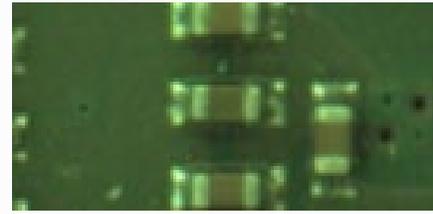
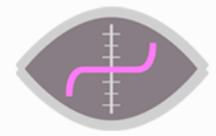
Обычная зона

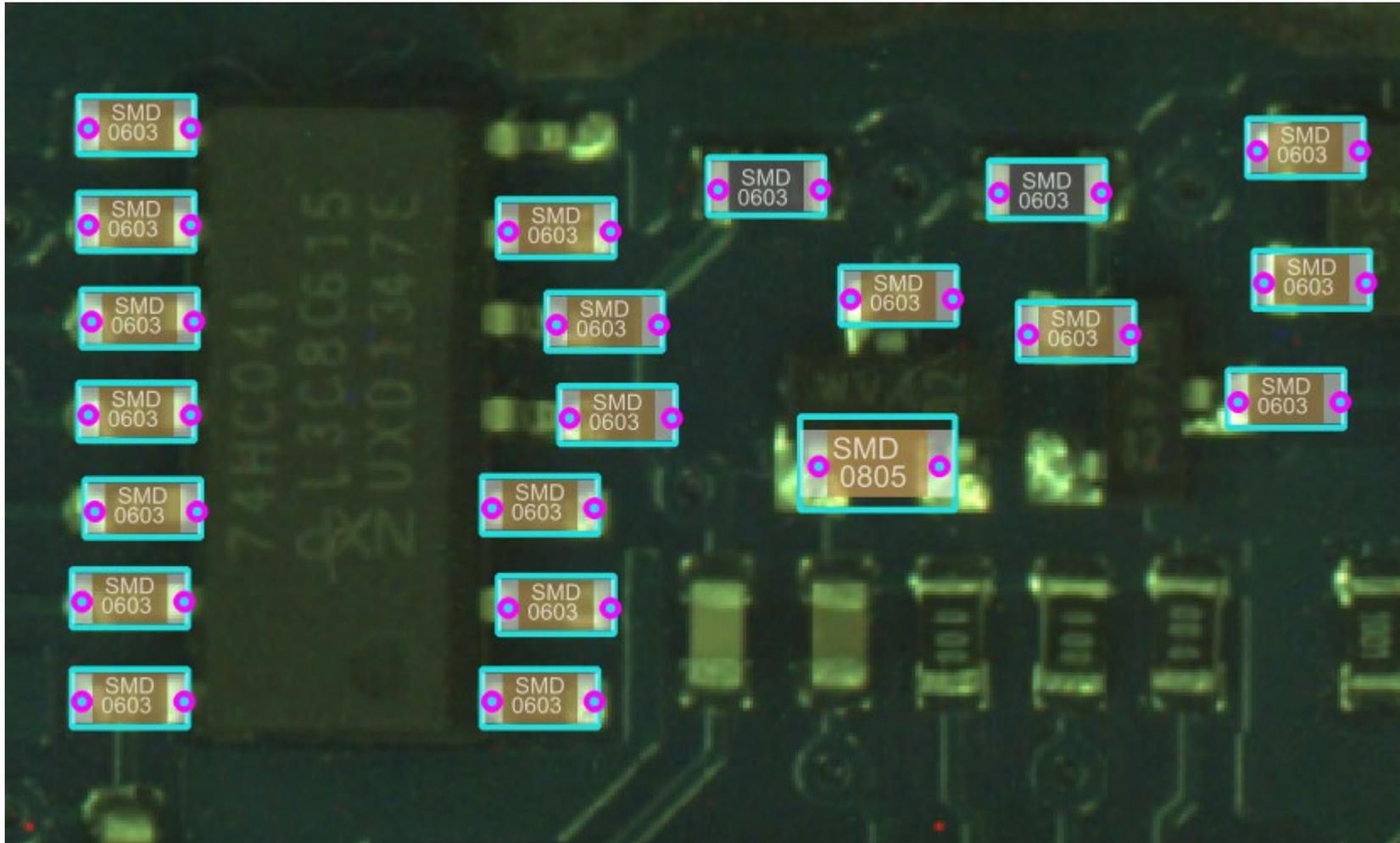
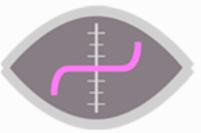


Увеличенная зона



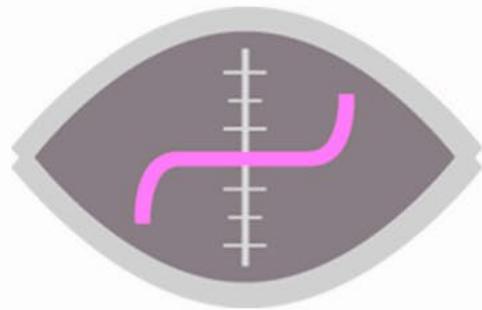
# Проверка гипотезы





Вероятными причинами неудачи могут быть:

- Извлечение паразитных фич из увеличенной зоны
- Недостаточная база чтобы сформировать закономерности в увеличенной зоне
- Неверный подбор параметров нейронной сети



# EyePoint P10

New look at troubleshooting  
with neural networks

Курсовая работа “EyePoint P10”  
19.05.2020

Выполнил:  
студент физического факультета МГУ  
Маракулин Андрей Павлович

Руководитель:  
кандидат физико-математических наук  
Запуниди Сергей Александрович