

Растеризованные
страницы документа

Построчное сканирование
страницы и классификация
каждой строки на основе
распределения пикселей
в ней

A21

Классифицированные
строки

Объединение строк в
сегменты на основе
правил перехода в
конечном автомате
состояний

A22

Классифицированные
сегменты

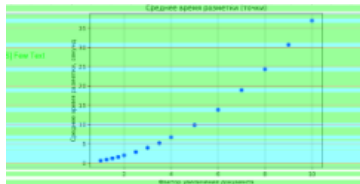


Рисунок 3 – Зависимость времени работы алгоритма от коэффициента
увеличения документа при его преобразовании в растровый формат

Алгоритм

Для определения зависимости времени работы от фактора увеличе-
ния документа алгоритм строит следующую аппроксимацию на доступных данных:

```
1) import numpy as np
2) x = np.arange(1 ... 10)
3) y = np.arange(1 ... 35)
4) log_x = np.log(x)
5) log_y = np.log(y)
6) % log_x = np.polyfit(log_x, log_y, 1)
7) x = np.exp(log_x)
8) print(f'Y_{log_x}(x) = 3.1 * x^{log_x(3.1)} > y = 0.583 * x^{1.792}
```

Полученная аппроксимация

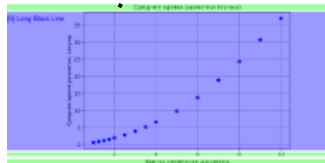


Рисунок 5 – Зависимость времени работы алгоритма от коэффициента
увеличения документа при его преобразовании в растровый формат

Алгоритм

Для определения зависимости времени работы от фактора увеличе-
ния документа алгоритм строит следующую аппроксимацию на доступных данных:

```
1) import numpy as np
2) x = np.arange(1 ... 10)
3) y = np.arange(1 ... 35)
4) log_x = np.log(x)
5) log_y = np.log(y)
6) % log_x = np.polyfit(log_x, log_y, 1)
7) x = np.exp(log_x)
8) print(f'Y_{log_x}(x) = 3.1 * x^{log_x(3.1)} > y = 0.583 * x^{1.792}
```

Полученная аппроксимация