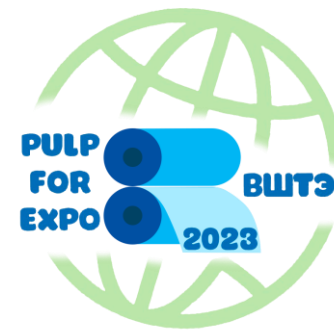


ЭКСПРЕСС АНАЛИЗ КАЧЕСТВА ОТБЕЛКИ БУМАЖНОГО ПОЛОТНА НА БАЗЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА



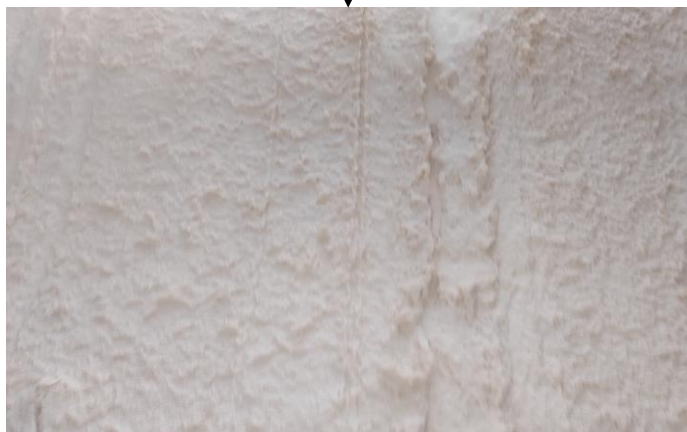
Анисимов Илья, студент 529 группы
ВШТЭ, СПб | 13 - 14. 11. 2023



Отбелка целлюлозы

Современная целлюлозно-бумажная промышленность модифицирует свои технологии отбелки при производстве целлюлозы для повышения качества конечной продукции и снижения затрат на реагенты, при этом одной из задач является оперативный контроль качества отбелки.

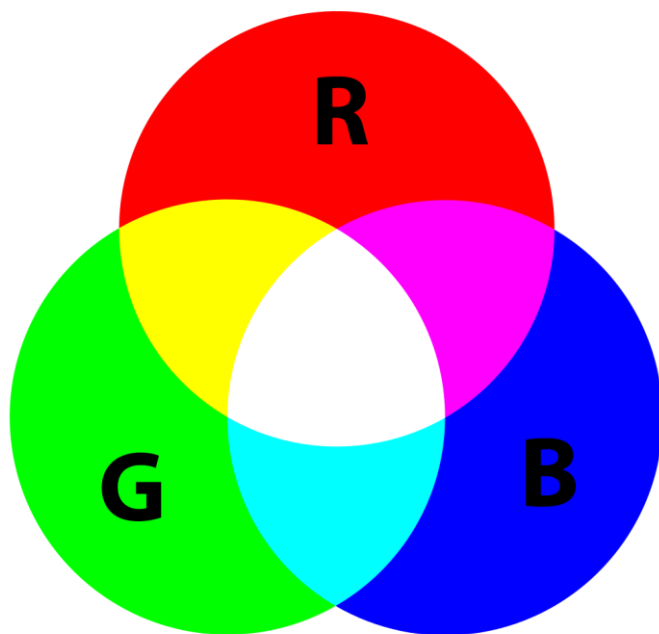




Первичная обработка снимков

- Устранение областей с пересветом
- Устранение затемненных участков
- Удаление снимков с различными дефектами



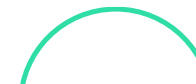


Относительная яркость

Для получения относительной яркости из фотографии необходимо преобразовать цвета пикселя по формуле:

$$L = 0,2126 * R + 0,7152 * G + 0,0722 * B$$

R, G, B – коды красной, зеленой, голубой компоненты цвета в цветовом пространстве RGB



Проверка правильности выбора связываемых параметров

Белизна

Открыть Преобразовать Сохранить Открыть Преобразовать

Средняя яркость

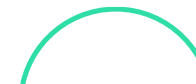
Белизна

Средняя яркость

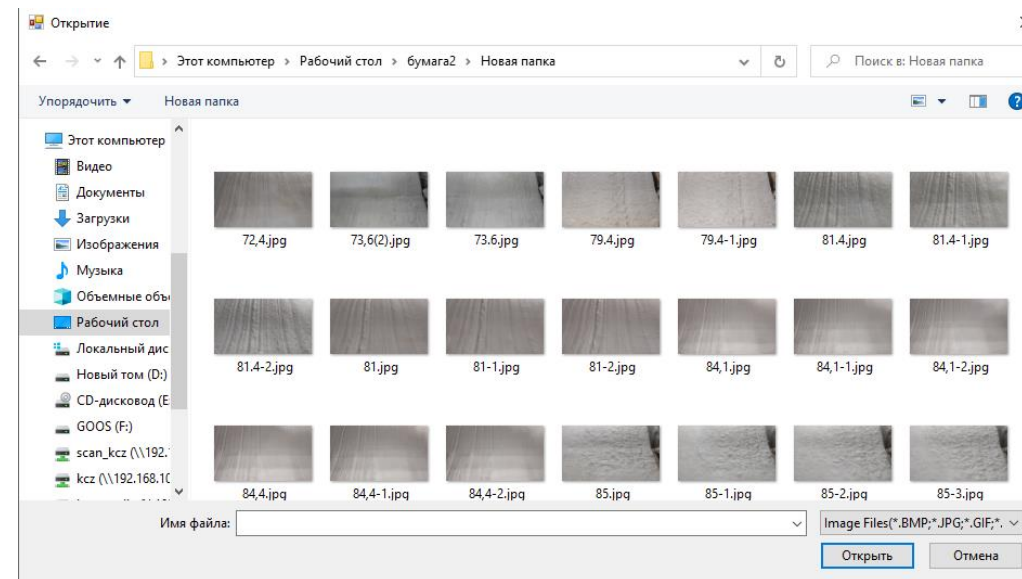
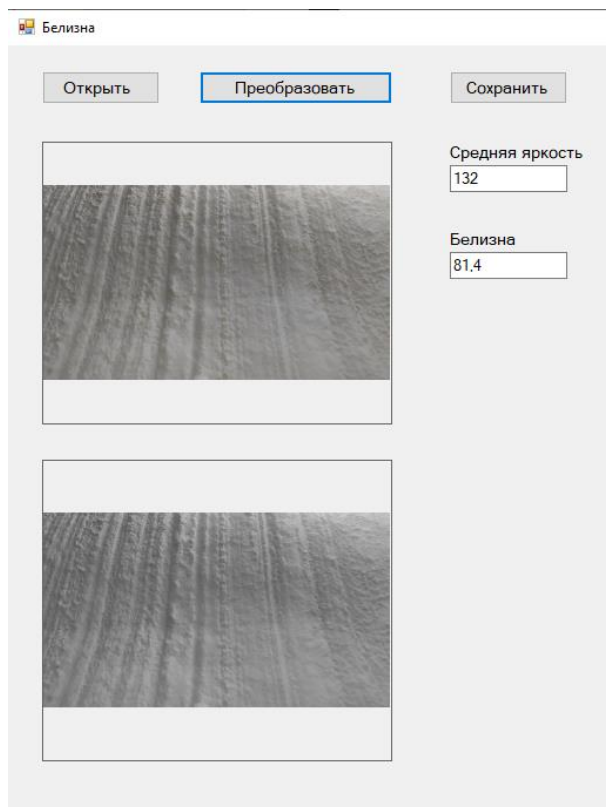
Белизна

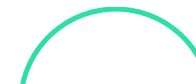
Сбросить



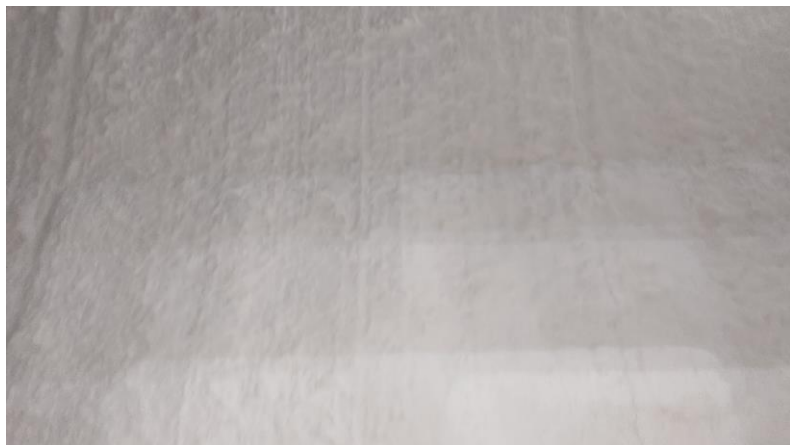


Пример работы с программой





Результат работы программы



Фотография целлюлозного
полотна с известной
белизной – 86.9%

Открыть Преобразовать

Средняя яркость
165

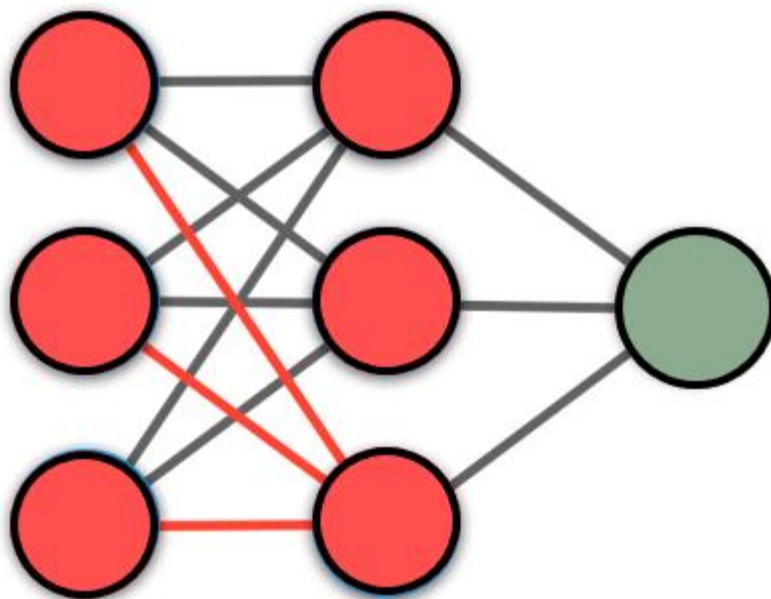
Белизна
79.4

Сбросить



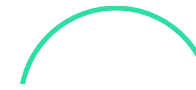


Применение нейронной сети

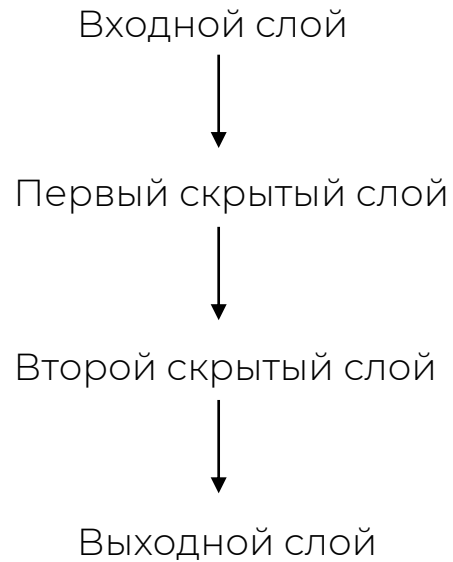


- Прямое распространение ошибки
- 2 полносвязных слоя
- 3 входных нейрона
- Сигмоидная функция активации





Алгоритм работы нейронной сети



1. Передача входных данных в нейронную сеть
2. Вычисляется взвешенная сумма входов для каждого нейрона в скрытом слое,
Применяется сигмоидная функция активации
3. Вычисляется взвешенная сумма входов для каждого нейрона в скрытом слое,
Применяется сигмоидная функция активации
4. Вычисляется взвешенная сумма входов для каждого нейрона в выходном слое,
Применяется сигмоидная функция активации



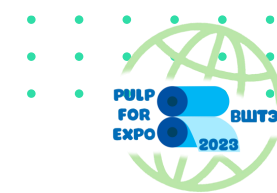
Результат работы нейронной сети

```
# Задаем обучающие данные
inputs = np.array([[0.031929907, 0.008995767, 0.103842958], [0.0340
[0.032391864, 0.009182964, 0.112287245], [0.031
[0.033509818, 0.010165863, 0.119623478], [0.032852
[0.031937433, 0.008968039, 0.095180081], [0.031340908
[0.030669748, 0.008367928, 0.080221797], [0.0305
[0.03179388, 0.008882302, 0.09161612], [0.03153585, 0
[0.0307822, 0.00851523, 0.0832753], [0.03104378, 0.00
[0.03113642, 0.008561973, 0.08349002], [0.0321963, 0.
[0.02970437, 0.008294579, 0.07814436], [0.03093799, 0.
[0.03076404, 0.008364378, 0.0819139], [0.03016501, 0.
[0.03183466, 0.008814236, 0.09887964], [0.03052665, 0
[0.03084545, 0.008328472, 0.07923607], [0.03062565, 0
outputs = np.array([[0.013927577, 0.013812155, 0.013157895, 0.012804097,
0.011560694, 0.011520737, 0.011494253, 0.011848341
0.01150748, 0.013245033, 0.011587486, 0.01305483, 0
0.011933174, 0.01150748, 0.012820513, 0.012019231,
```

Результат при загрузке
фотографии с известной
белизной – 71,7%

Prediction for [1, 0, 0]: [0.01233332]

	расчетное значение белизны, %			
1/расч	81,0812			



Спасибо за внимание !

