

**Алгоритм обработки
изображений деталей для их
классификации**

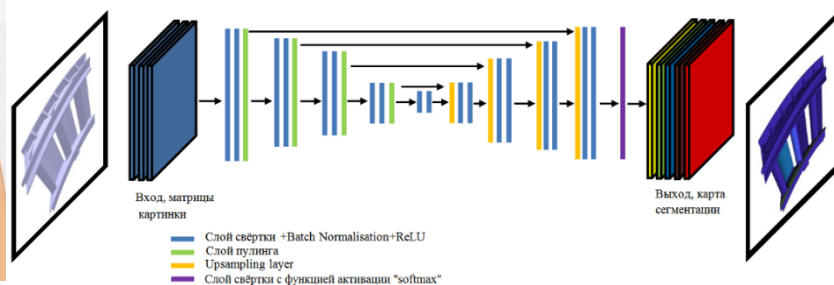
***Component Image Processing
Algorithm for Classification***

Е.Ю. Печенина, В.А. Печенин и
М.А. Болотов

Самарский университет, г. Самара,
Российская Федерация, ек-ko[@list.ru](mailto:ek-ko@list.ru)

Цель работы


- Одним из направлений повышения качества, эффективности и снижения расходов производства является систематическая цифровая трансформация существующих процессов.
- Информация по деталям храниться и обрабатывается в электронном виде.
- Для отслеживания деталей в производстве эффективным решением являются системы компьютерного зрения, основанные на использовании методов машинного обучения.



В работе рассматривается проблема создания надежного алгоритма обработки данных для повышения надежности классификации деталей аэрокосмических предприятий.

Этапы алгоритма обработки изображений

- Идея состоит в обучении нейронных сетей по STL моделям.
- Распознавание же будет осуществляться на реальных изображениях.
- Создан алгоритм подготовки данных к виду, похожему на получаемые изображения с stl-моделей



```
graph LR; A[Билатеральная фильтрация для удаления шумов] --> B[Проведение пороговой обработки]; B --> C[Наложение полученной монохромной маски на исходное изображение]; C --> D[Преобразование в оттенки серого];
```

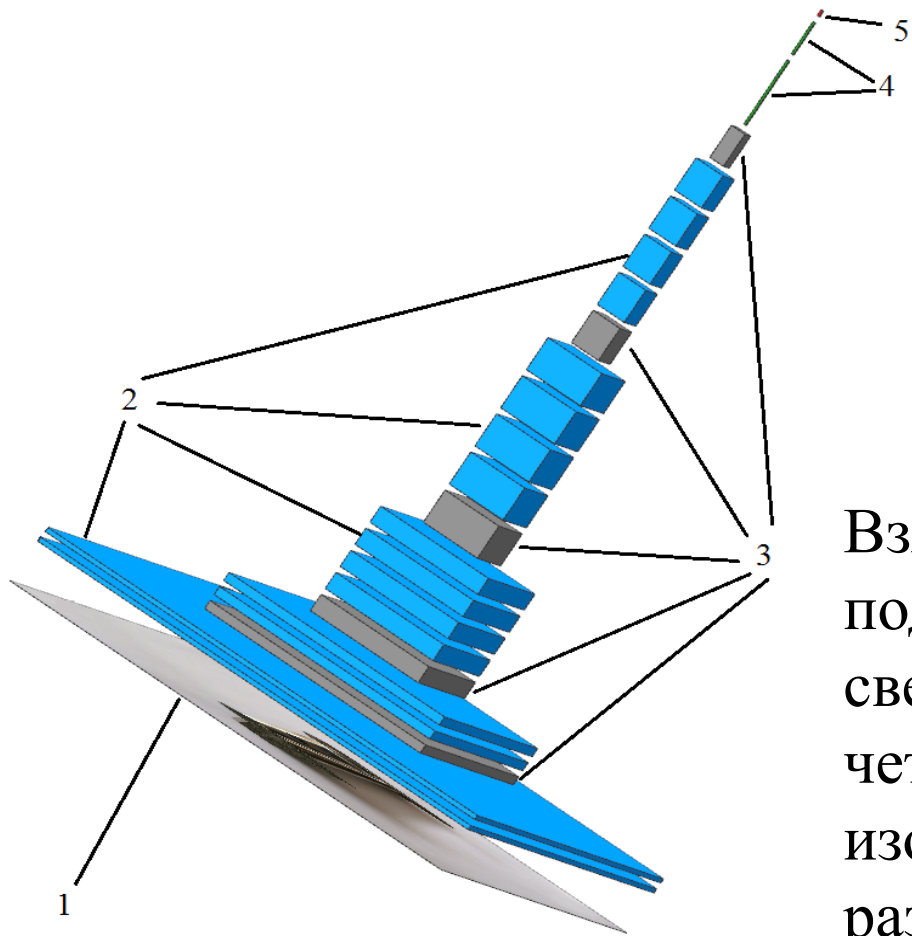
Билатеральная
фильтрация для
удаления
шумов

Проведение
пороговой
обработки

Наложение полученной
монохромной маски на
исходное изображение

Преобразование
в оттенки серого

Архитектура нейронной сети



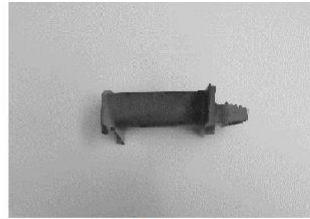
Архитектура сети VGG 1) Входное изображение; 2) сверточные слой с функцией активации ReLu; 3) max pooling слой; 4) полносвязанные слои с ReLu; 5) выходной слой с функцией активации softmax

Взято несколько вариантов VGG-подобной модели: с одним сверточным слоем, двумя, тремя и четырьмя. На вход подается изображение в «оттенках серого» размером 112×112 пикселей (первые два варианта архитектуры) и 224×224 пикселя (третий и четвертый варианты). Значения пикселей нормированы в диапазоне от 0 до 1.

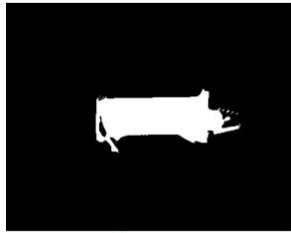
Результаты исследования



a



b

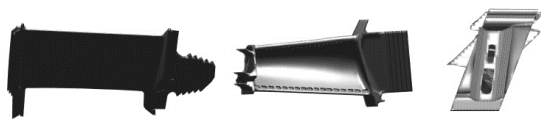


c

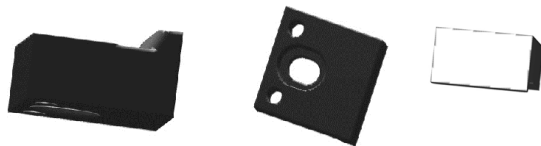


d

Результаты работы алгоритма обработки а) исходное изображение б) билатеральная фильтрация; с) пороговая обработка д) выделенный объект



a



b

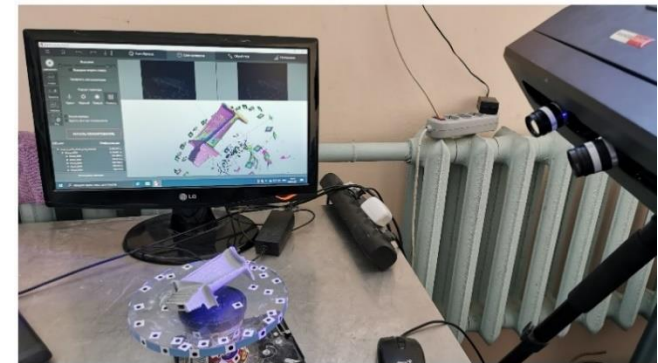
Проекции STL моделей а) турбинная лопатка; б) деталь для фиксации.



a



c



b

Данные для классификации а) STL модель; б) процесс сканирования с) фотография детали.

Точность δ_{class} и функция потерь L_f при классификации

$$\delta_{class} = N_{corr_c} / N_{total}$$

Архитектура	Архитектура 1		Архитектура 2		Архитектура 3		Архитектура 4	
	μ	σ	μ	σ	μ	σ	μ	σ
Обучающая	0,9912	0,0137	0,8824	0,1833	0,7383	0,2055	0,6911	0,1748
Проверочная	0,9681	0,0260	0,8731	0,1747	0,7420	0,2027	0,6796	0,1804
Без обработки	0,7870	0,3951	0,8148	0,3016	0,3519	0,4108	0,7407	0,3779
Обработанные	0,2222	0,2018	0,5093	0,3573	0,6481	0,4241	0,8148	0,3056

L_f - категориальная перекрестная энтропия

Архитектура	Архитектура 1		Архитектура 2		Архитектура 3		Архитектура 4	
	μ	σ	μ	σ	μ	σ	μ	σ
Обучающая	0,0257	0,0329	0,9186	1,8390	2,4145	3,0908	3,2499	3,2107
Проверочная	0,0893	0,0743	0,9397	1,7755	2,3605	2,9889	3,3520	3,3400
Без обработки	1,9720	4,4213	1,6855	2,3313	6,6282	5,4640	1,5384	1,8711
Обработанные	2,8120	1,0305	1,8696	1,7100	1,4348	1,8600	0,8651	1,4636

Величина функции потерь предсказуемо растет с увеличением количества слоев, что говорит о необходимости большего количества данных для обучения. В то же время для тестовых фотографий, подвергнутых обработки по предложенному алгоритму, снижается.

Выводы

- Предложенный в статье алгоритм обработки фотографий позволяет повышать стабильность классификации.
- Предложенный в статье алгоритм обработки фотографий позволяет повышать стабильность классификации.
- Дальнейшими направлениями исследованиями являются эксперименты с фильтрацией шумов на изображениях, использованием морфологических операций над изображениями, а также подбор иных архитектур нейронных сетей.