# ТЕХНИЧЕСКИЕ НАУКИ



УДК 004.62

## Обучающая выборка для сегментации изображений сельскохозяйственных культур

**И.М. Благородов, А.В. Шрамченко**
Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

**Аннотация**
Рассматриваются методы подготовки обучающих выборок данных для сегментации изображений сельскохозяйственных культур, представляющих собой посевные площади. Эти выборки явились основой для создания синтетических данных, используемых для обучения разрабатываемой системы распознавания и классификации участков посевных площадей, где отсутствуют всходы или наблюдается наличие сорняков. В качестве исследуемой культуры выбрана пшеница. Семантическая сегментация сорняков на изображениях, а также их полигональное определение осуществляется с помощью веб-платформы для аннотирования изображений Cvat. В статье представлены программные скрипты для подготовки обучающей выборки для сегментации изображений сельскохозяйственных культур, написанные на языках Matlab и Python.

**Ключевые слова:** сегментация изображений, сельскохозяйственные культуры, компьютерное зрение, синтетическая выборка, генерация синтетической выборки, Matlab, Python, OpenCV

**Для цитирования.** Благородов И.М., Шрамченко А.В. Обучающая выборка для сегментации изображений сельскохозяйственных культур. *Молодой исследователь Дона*. 2025;10(3):60–64.

## A Training Sample for Crop Image Segmentation

**Igor M. Blagorodov, Aleksandr V. Shramchenko**
Don State Technical University, Rostov-on-Don, Russian Federation

**Abstract**
Training sample preparing techniques for segmenting images of agricultural crops on the cultivated lands were studied. Such samples served the basis for creating the synthetic data to train a system developed for recognition and classification of the cultivated lands without seedlings or with weeds. Wheat was chosen as the crop to study. Semantic image segmentation of weeds, including their detection with polygon annotation, were carried out using CVAT — image and video data annotation platform. The article presents the programming scripts written using Matlab with Python for preparing a training sample for crop image segmentation.

**Keywords:** image segmentation, agricultural crops, computer vision, synthetic sample, synthetic data generation, Matlab, Python, OpenCV

**For Citation.** Blagorodov IM, Shramchenko AV. A Training Sample for Crop Image Segmentation. *Young Researcher of Don.* 2025;10(3):60–64.

**Введение.** Основным трендом в области мониторинга состояния посевных площадей стало применение автоматизированных систем сбора данных. Они представлены роботизированными платформами, беспилотными летательными аппаратами, стационарными метеостанциями, а также системами автоматизации, монтируемыми на сельскохозяйственную технику. Последние все больше привлекают внимание за счет своей универсальности, модульности и относительно низкой стоимости, и именно такой тип систем сбора данных рассматривается в качестве аппаратной основы в статье.

Визуальный осмотр полей трансформируется в процедуру обработки изображений системами технического зрения, что значительно повышает эффективность трудоемкого процесса оценки сельскохозяйственных ландшафтов. Разработка систем технического зрения обладает практической ценностью, так как значительно упрощает, ускоряет и автоматизирует сбор и обработку информации о состоянии посевных площадей на всех этапах выращивания сельскохозяйственных культур. Однако существует ряд ограничений и особенностей, накладываемых на реальные данные или датасеты в открытом доступе, которые негативно сказываются на качестве моделей

машинного обучения. В первую очередь, это затрудненность процесса получения качественных изображений в достаточном количестве из-за погодных условий, а также зависимость сбора данных от временных мероприятий по подготовке посевных площадей к высеву и их последующему уходу. Кроме того, некоторые виды растений или исследуемые характеристики состояния посевных площадей встречаются реже при сборе данных, что, в конечном итоге, приводит к дисбалансу классов при машинном обучении. В этой связи актуальной задачей является генерация синтетических данных, которая позволит создать сбалансированный набор данных, где каждый класс представлен достаточным количеством примеров для корректного обучения [1–3]. Цель работы заключается в повышении эффективности машинного обучения систем оценки посевных площадей на основе технического зрения с помощью создания сбалансированного набора данных о сорности поля.

**Основная часть.** Данные для нашего проекта были получены из датасетов, представленных на сайте «Kaggle», который является популярной платформой для обмена данными и решения задач в области анализа данных. В рамках нашего исследования была проведена выборка изображений сельскохозяйственных культур, снятых сверху, с различимым грунтом между рядами и наличием сорняков. В частности, в статье рассматривается генерация синтетической выборки для пшеницы. Первичная выборка состояла из 500 фотографий пшеницы, из которых 120 фотографий включали в себя различные виды сорняков [4–9].

После получения первичной выборки данных необходимо провести обработку изображений и обучение модели компьютерного зрения. Основная задача заключалась в том, чтобы модель могла эффективно различать урожай и сорняки на изображениях. Для достижения этой цели необходимо было сегментировать данные по типу растений. Для этого использовалась онлайн платформа CVAT (Computer Vision Annotation Tool), которая является мощным инструментом для аннотирования изображений и видео. В качестве метода сегментации применялась семантическая сегментация с выделением на изображении полигонов сорняков и их автоматическим аннотированием (рис. 1).

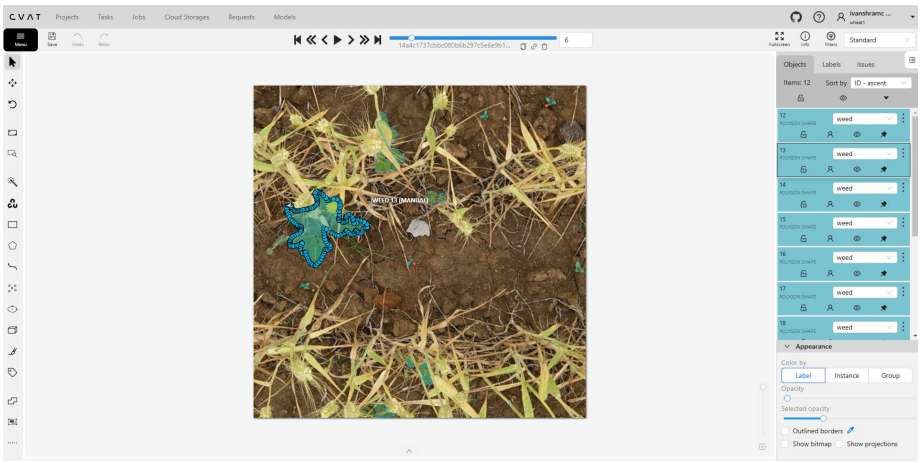


Рис. 1. Сегментация изображения

В ходе работы было обработано 500 изображений из выбранных датасетов, что дало возможность получить 120 аннотаций (рис. 2) с 300 полигональными сегментами сорняков для тренировки модели.

```
<annotations>
  <version>1.1</version>
  <meta>
    <job>
      <id>2025176</id>
      <size>1</size>
      <mode>annotation</mode>
      <overlap>0</overlap>
      <bugtracker/>
      <created>2025-02-10 13:07:49.695296+00:00</created>
      <updated>2025-02-17 19:47:56.198147+00:00</updated>
      <subset>default</subset>
      <start_frame>33</start_frame>
      <stop_frame>33</stop_frame>
      <frame_filter/>
      <segments>
      ...
      </segments>
      <owner>
      ...
      </owner>
      <assignee/>
      <labels>
      ...
      </labels>
    </job>
    <dumped>2025-02-17 19:49:21.835087+00:00</dumped>
  </meta>
  <image id="33" name="фото культуры с сорняками/afacbdc35674944c59e01a339798dace58a7b239d812eddf01cc3015e4b165d5.jpg" width="1024" height="1024">
    <polygon label="weeds" source="semi-auto" occluded="0"
    points="383.00,0.00;381.00,15.00;376.00,25.00;376.00,37.00;382.00,51.00;383.00,59.00;386.00,63.00;401.00,65.00;410.00,73.00;415.00,73.00;420.00,59
    </polygon>
    <polygon label="weeds" source="semi-auto" occluded="0"
    points="161.00,519.00;161.00,526.00;165.00,535.00;174.00,540.00;181.00,551.00;187.00,551.00;195.00,547.00;213.00,550.00;228.00,535.00;242.00,535.0
    z_order="0"> </polygon>
  </image>
</annotations>
```

Рис. 2. Аннотация изображения с полигональным сегментированием

Для выполнения задачи с генерацией синтетической выборки было принято решение рассмотреть два метода обработки изображений, которые позволят добиться необходимых результатов исследования.

Первый метод включает в себя использование программной среды для численных вычислений, визуализации данных и обработки изображений MATLAB, которая достаточно широко используется в научных, инженерных и исследовательских областях. Второй метод основан на использовании скриптов Python с OpenCV. Рассмотрим некоторые преимущества и недостатки обоих методов (Таблица 1).

Таблица 1

**Преимущества и недостатки используемых методов**

| Метод | Преимущества | Недостатки |
|---|---|---|
| MATLAB-scripts | Специальная функция для сегментации изображения: «imcrop», простота визуализации высококачественных изображений в режиме live-script | Невозможно реализовать сегментирование полигональных объектов без привлечения инструментов интерактивной полигональной рисовки и маскирования |
| Python+ OpenCV | Python позволяет легко интегрировать OpenCV с другими библиотеками, такими как «NumPy» для численных расчетов и «Matplotlib» для визуализации | Невозможно оптимизировать используемые библиотеки функций без глубокого погружения в программную разработку |

Выделенные особенности использования методов достаточно субъективны и решающей характеристикой является скорость обработки данных. Листинг скриптов для сравнения скорости обработки представлен в таблице 2.

Таблица 2

**Листинг скриптов для сравнения скорости обработки**

MATLAB-script

```
addpath ('C:\.....);
filename = ('1 (28).jpg')
annot = xmlread('annotations.xml')
I = imread(filename);
polygon(:,2) = polygon(:,2);
polygon = int32(polygon)
[n, m, k] = size(I);
Imasked = zeros(n,m,k,class(I));
BW = roipoly(I,polygon(:,1),polygon(:,2));
imshow(I)
figure
imshow(BW)
for i = 1:n
  for j = 1:m
    if BW(i,j) ==1
      Imasked(i,j,:) = I(i,j,:);
    end
  end
end
figure
imshow(Imasked)
imwrite(Imasked,'C:...\picture1.png');
imwrite(I,'C:...picture0.png');
Ipic = imread('C:...\picture1.png');
addpath ('C:...');
X=imread('picture1.png');
Y=imread('picture1.png');
i=size(X,1);
j=size(X,2);
C=uint8(zeros(i,j));
```

Python-script

```
import cv2
import xml.etree.ElementTree as ET
import numpy as np
import os
# Путь к XML-файлу (замени на свой путь)
xml_file = "C: ... /annotations.xml"
# Проверяем существование файла
if not os.path.exists(xml_file):
  print(f"Ошибка: файл {xml_file} не найден!")
  exit()
# Парсим XML
tree = ET.parse(xml_file)
root = tree.getroot()

# Создаем выходную папку
output_folder = "weed"
os.makedirs(output_folder, exist_ok=True)
# Обрабатываем каждое изображение в XML
for image_elem in root.findall("image"):
  image_name = image_elem.get("name")
  width = int(image_elem.get("width"))
  height = int(image_elem.get("height"))
  # Проверяем, есть ли полигоны
  polygons = image_elem.findall("polygon")
  if not polygons:
    print(f"Пропускаем {image_name} (нет аннотаций)")
    continue

  print(f"Обрабатываю {image_name} ({len(polygons)}
полигонов)")
```

```
for i1=1:1:i
  for j1=1:1:j
    if X(i1,j1)==255

     C(i1,j1)=0;
    end

  end
end
for i1=1:1:i
  for j1=1:1:j
    if X(i1,j1)~=255

     C(i1,j1)=X(i1,j1)*100;
    end
  end
end
imwrite (Y,'C:...ps.png','Alpha',C);
```

```
  # Загружаем изображение
  image = cv2.imread(image_name, cv2.IMREAD_UN-
CHANGED)
  if image is None:
    print(f"Ошибка: не удалось загрузить
{image_name}")
    continue
  # Обрабатываем каждый полигон отдельно
  for idx, polygon_elem in enumerate(polygons):
    points_str = polygon_elem.get("points")
    points = np.array([list(map(float, p.split(','))) for p in
points_str.split(';')], np.int32)
    # Создаем пустую маску для каждого полигона
    mask = np.zeros((height, width), dtype=np.uint8)
    cv2.fillPoly(mask, [points], 255)
    # Применяем маску к изображению
    result = cv2.bitwise_and(image, image, mask=mask)
    # Создаем альфа-канал (делаем фон прозрачным)
    b, g, r = cv2.split(result)
    alpha_channel = np.ones(b.shape, dtype=b.dtype) * 255
    alpha_channel[mask == 0] = 0
    # Объединяем каналы
    result_with_alpha = cv2.merge((b, g, r, alpha_channel))
```

При сравнении времени исполнения скриптов было установлено, что Matlab справляется с выделением одного сегмента из изображений и сохранением их в новый файл за 3,406 с. Использование скрипта Python позволяет решить аналогичную задачу за 1,40 с. (Таблица 2). Таким образом, в рассматриваемой задаче использование Python-scripts является наилучшим решением для обработки больших данных.

**Заключение.** Применение современных программных решений и технологий, таких как компьютерное зрение, открывает новые перспективы для автоматизации мониторинга и анализа посевных участков. Это не только способствует повышению эффективности управления сельским хозяйством, но и позволяет более точно оценивать состояние культур, оптимизируя процессы ухода и сбора урожая. Внедрение таких технологий улучшает как урожайность, так и устойчивость сельского хозяйства, что является ключевым аспектом для обеспечения продовольственной безопасности.

В рамках данного исследования была решена задача генерации синтетических данных с использованием программ MATLAB и Python с библиотекой OpenCV. Авторы сравнили их производительность и актуальность для данной задачи и пришли к выводу, что Python с OpenCV представляет собой более подходящий вариант, если нужна гибкость и возможность интеграции с другими библиотеками. Этот выбор особенно подходит для разработчиков, стремящихся создать масштабируемые и расширяемые решения в области компьютерного зрения и машинного обучения.

**Финансирование.** Работа проведена в рамках выполнения проекта «Математическое моделирование и алгоритмы моделирования роста растений на основе автоматизированной картографической системы» (FZNE2024-0006).

**Список литературы:**

1. Schneider F, Swiatek J, Jelali M. Detection of Growth Stages of Chilli Plants in a Hydroponic Grower Using Machine Vision And YOLOv8 Deep Learning Algorithms. *Sustainability.* 2024;16(15):6420. https://doi.org/10.3390/su16156420

2. Liu S, Qi L, Qin H, Shi J, Jia J. Path Aggregation Network for Instance Segmentation. In: *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE; 2018. P. 8759–8768. https://doi.org/10.1109/CVPR.2018.00913

3. Ajadi AO, Barr J, Liang SZ, Ferreira R, Kumpatla SP. Large-Scale Crop Type and Crop Area Mapping across Brazil Using Synthetic Aperture Radar and Optical Imagery. *International Journal of Applied Earth Observation and Geoinformation*. 2021;97: 102294. https://doi.org/10.1016/j.jag.2020.102294

4. *Global Wheat Head Dataset. Kaggle: Your Machine Learning and Data Science Community; 2024.* URL: https://www.kaggle.com/datasets/preetam009/global-wheat-head-dataset?select=utokyo_2 (accessed: 29.01.2025).

5. *Global Wheat WILDS V1.1. Kaggle: Your Machine Learning and Data Science Community; 2022.* URL: https://www.kaggle.com/datasets/waskita/global-wheat-v-1-1 (accessed: 29.01.2025).

6. *Global Wheat Challenge 2021. Kaggle: Your Machine Learning and Data Science Community; 2021.* URL: https://www.kaggle.com/datasets/bendvd/global-wheat-challenge-2021 (accessed: 30.01.2025).

7. *Global Wheat Challenge. Kaggle: Your Machine Learning and Data Science Community; 2021.* URL: https://www.kaggle.com/datasets/ipythonx/global-wheat-challenge (accessed: 30.01.2025).

8. *Global Wheat Head Dataset (Manually Refined). Kaggle: Your Machine Learning and Data Science Community; 2020.* URL: https://www.kaggle.com/datasets/matthewmasters/global-wheat-head-dataset-manually-refined (accessed: 30.01.2025).

9. *Global Wheat Head Dataset. Kaggle: Your Machine Learning and Data Science Community; 2023.* URL: https://www.kaggle.com/datasets/preetam009/global-wheat-head-dataset (accessed: 31.01.2025).

***Об авторах:***

**Игорь Михайлович Благородов,** студент кафедры автоматизации производственных процессов Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), boss.blagorodov@mail.ru

**Александр Вячеславович Шрамченко,** студент кафедры автоматизации производственных процессов Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), aleksandr_shramchenko@mail.ru

***Конфликт интересов:*** **авторы заявляют об отсутствии конфликта интересов.**

***Все авторы прочитали и одобрили окончательный вариант рукописи.***

***About the Authors:***

**Igor M. Blagorodov,** Student of the Automation of Production Processes Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation), boss.blagorodov@mail.ru

**Alexander V. Shramchenko,** Student of the Automation of Production Processes Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation), aleksandr_shramchenko@mail.ru

***Conflict of Interest Statement:*** **the authors declare no conflict of interest.**

***All authors have read and approved the final manuscript.***