

## Робот для ремонта трещины в конструкции

*Д. В. Третьяков<sup>1\*</sup>, Р. И. Чижменко<sup>1</sup>, В. В. Паксутов<sup>1</sup>*

<sup>1</sup> Сибирский государственный университет путей сообщения, г. Новосибирск,  
Российская Федерация  
\* e-mail: Dim22037@gmail.com

**Аннотация.** На основе технологий машинного обучения и создания нейросетей представлен макет разработки робота для ремонта трещин в конструкции мостовых сооружений. На основе ранее проведенных исследований стало известно, что трещины в конструкции моста нужно именно запаивать, и обязательно при помощи робота.

**Ключевые слова:** мостовые конструкции, трещины, обработка фотоизображений, ремонт трещины металлоконструкций, распознавание объектов, CNN, UNet, концепция робота

## Robot for repairing a crack in a structure

*D. V. Tretyakov<sup>1\*</sup>, R. I. Chizhmenko<sup>1</sup>, V. V. Paksutov<sup>1</sup>*

<sup>1</sup> Siberian transport university, Novosibirsk, Russian Federation  
\* e-mail: Dim22037@gmail.com

**Abstract.** On the basis of machine learning technologies and the creation of neural networks, a mock-up of the development of a robot for repairing cracks in the structure of bridge structures is presented. Based on previous studies, it became known that cracks in the bridge structure need to be soldered, and always with the help of a robot.

**Keywords:** bridge constructions, cracks, photo image processing, repair of cracks in metal structures, object recognition, CNN, UNet, robot concept

### *Введение*

Трещины в конструкциях мостов – это одна из значимых проблем современного мостостроения. С недавнего времени было принято решение, что трещины в конструкциях мостов нужно запаивать, а не варить.

Актуальность проблемы связана с тем, что ручная пайка трещины запрещена, так как это нарушает требования и регламенты работы на пролетных строениях мостовых конструкций. Необходим робот, который сможет распознать координаты поврежденной области и провести запайку с нужной нам точностью (рис. 1).

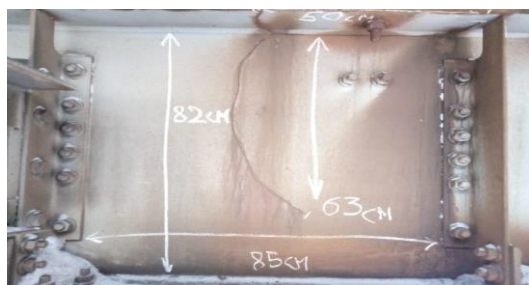


Рис. 1. Трещина в балке пролетного строения

## ***Цель работы***

Целью работы является разработка концепта робота по ремонту трещин пролетных строений металлоконструкций мостовых сооружений.

Для достижения этой цели необходимо решить следующие задачи.

1. Отредактировать изображение трещины.
2. Определить метод обработки изображения для вывода координат трещины.
3. Обработать координаты.
4. Написать программу, задающую траекторию движения робота.

## ***Редактирование изображения***

Обработаем изображение так, чтобы было удобнее искать координаты линии трещины и находить ее натуральную величину на плоскости.

Для этого используем программу Adobe Photoshop.

Открываем первоначальное изображение в программе

Выделяем нужную нам область при помощи команды «Выделение».

Обрезаем фото по выделенной области.

При помощи функции «Трансформация» растягиваем получившийся фрагмент для получения нужного нам разрешения файла.

Сохраняем получившееся изображение (рис. 2).

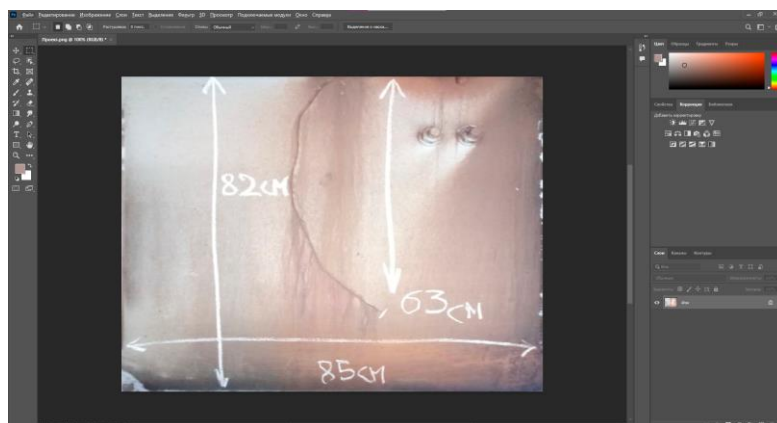


Рис. 2. Редактирование изображения

## ***Сегментация изображения с помощью СНС на архитектуре UNet***

В последние годы значительное число исследований в области поиска и выделения объектов на изображениях было сосредоточено на применении нейросетевого подхода к распознаванию, данный подход показал наибольшую эффективность по сравнению с другими процедурами. Применение при распознавании объектов нейронных сетей требует меньшего числа настроек и хорошо подходит для решения широкого спектра задач [1].

В ходе изучения сверточных нейронных сетей (СНС) было принято решение по выбору нейронной сети на архитектуре UNet, так как модель UNet часто используется для сегментации медицинских изображений [2], где выполняет поиск

аномалий в тканях и внутренних органах человека [3], и для обработки спутниковых фотоизображений с целью выделения крупных объектов, что приблизительно схоже с поставленной перед нами задачей по сегментации и выделению трещины в пролетном строении мостовых конструкций.

Нейронная сеть UNet, созданная для решения задачи семантической сегментации, делится на нисходящую (шифровальщик) и восходящую (де-шифровальщик) части (рис. 3). Шифровальщик состоит из повторного применения двух сверток  $3 \times 3$ , за которыми следуют активация ReLU и операция субдискретизации  $2 \times 2$  степени 2 для понижения разрешения. На каждом этапе понижающей дискретизации каналы свойств удваиваются. Для первичной инициализации сверточных слоев шифровальщика используются параметры, полученные путем применения алгоритма Kaiming. Каждый шаг в де-шифровальщике состоит из операции повышающей дискретизации карты свойств, за которой следуют: свертка  $2 \times 2$ , которая уменьшает количество каналов свойств; объединение с соответствующим образом обрезанной картой свойств из шифровальщика; две свертки  $3 \times 3$ , за которыми следует ReLU [4].



Рис. 3. Архитектура UNet

### *Решение задачи выявления трещины*

Рассмотрим решение задачи поиска повреждений (трещин) на изображениях пролетных строений металлоконструкций мостовых сооружений. Для решения задачи поиска повреждений предложено использовать глубокую сверточную нейронную сеть (CNN) с архитектурой UNet, которая позволяет определить не только класс изображения целиком, но и сегментировать его области по классу, т. е. создать маску, которая будет разделять изображение на несколько классов [4]. Архитектура состоит из стягивающего пути для захвата контекста и симметричного расширяющегося пути, который позволяет осуществить точную локализацию.

Начальной выборкой послужило несколько сотен фотографий, содержащих различные дефекты в элементах металлических конструкций. Для организации процесса обучения нейронной сети, с целью проведения последующей валидации, все имеющиеся в наличии фотографии были поделены на обучающую и тестовую выборки (в соотношении 80 % – 20 %). Была выполнена разметка трещин

в виде бинарных изображений, на которых трещины выделены белым цветом, а фон – черным (рис. 4).



Рис. 4. Пример обучающей модели

Благодаря обученной нейросети сегментируем и выявляем трещину на нашей первоначальной фотографии, после чего выводим ее координаты относительно заданного начала координат на изображении.

Полученные координаты переносим в MS Excel и строим график на основании полученных координат X и Y (рис. 5).

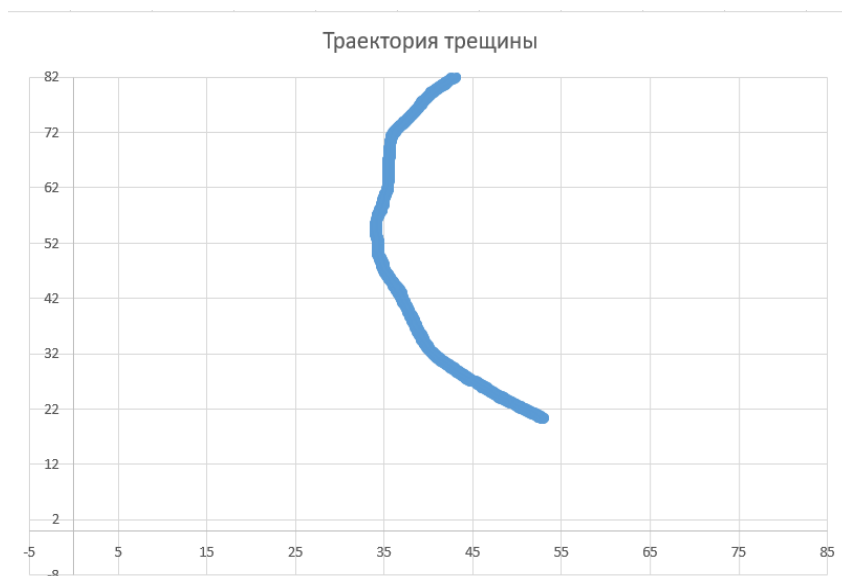


Рис. 5. График координат трещины в MS Excel

Проверяем правильность вывода координат относительно изображения трещины в пролетном строении, выведенной нейросетью (CNN) на языке Python, на архитектуре UNet. После того, как произведена проверка и не обнаружены лишние элементы на графике помимо трещины, приступаем к созданию концепции робота по ремонту трещины в пролетном строении металлоконструкции мостового полотна.

## **Концепт работы робота по ремонту трещины в пролетном строении металлоконструкции мостового полотна**

Из теоретической физики имеет выражение для вычисления сопротивления проводника

$$R = \frac{\rho}{S} L ,$$

где  $L$  – длина проводника.

Для однородного стержня постоянного сечения имеем:  $R \sim L; U \sim R$

При постоянной силе тока генератора

$$U(R) \sim L \quad (1)$$

То есть, каждая точка проводника имеет определенное значение напряжения [6].

Как было отмечено выше, разрешённым способом ремонта повреждённого полотна элемента пролетного строения является пайка. Для обеспечения максимальной прочности запаянного элемента, паять необходимо нанося припой продольно трещине. Исходя из этого требования траектория движения робота будет описываться векторами. Начало каждого вектора будет на одной границе трещины, конец на противоположной.

Иначе, для робота необходимо задать систему координат, соответствующей системе координат изображения.

Для реализации воспользуемся (2). Используя вольтметры, измеряющие напряжение на направляющих, отвечающих за перемещение по вертикали и горизонтали, получим для каждой точки соответствующие значения  $(U_x; U_y)$ .

Посредством программы на языке Python из общего массива точек оставляем только краевые значения. Полученные координаты являются границами трещины (рис. 6).

Для соответствия значений точек с координатами  $(X; Y)$  и  $(U_x; U_y)$ , используя, опять же, язык Python конвертируем значения  $X$  в соответствующие значения  $U_x$ , аналогично с  $Y$  и  $U_y$ .

Получаем:  $U_x = K \cdot X; U_y = K \cdot Y$  где  $K$  – коэффициент, определяющийся при проектировании робота.

Полученные координаты переносим в MS Excel.

Основой работы выступают модули Arduino, в качестве микроконтроллера будет использован Arduino Mega, так как он обладает наибольшей производительностью из всех имеющихся моделей.

Программирование микроконтроллера будет осуществляться на языке Arduino IDE [5].

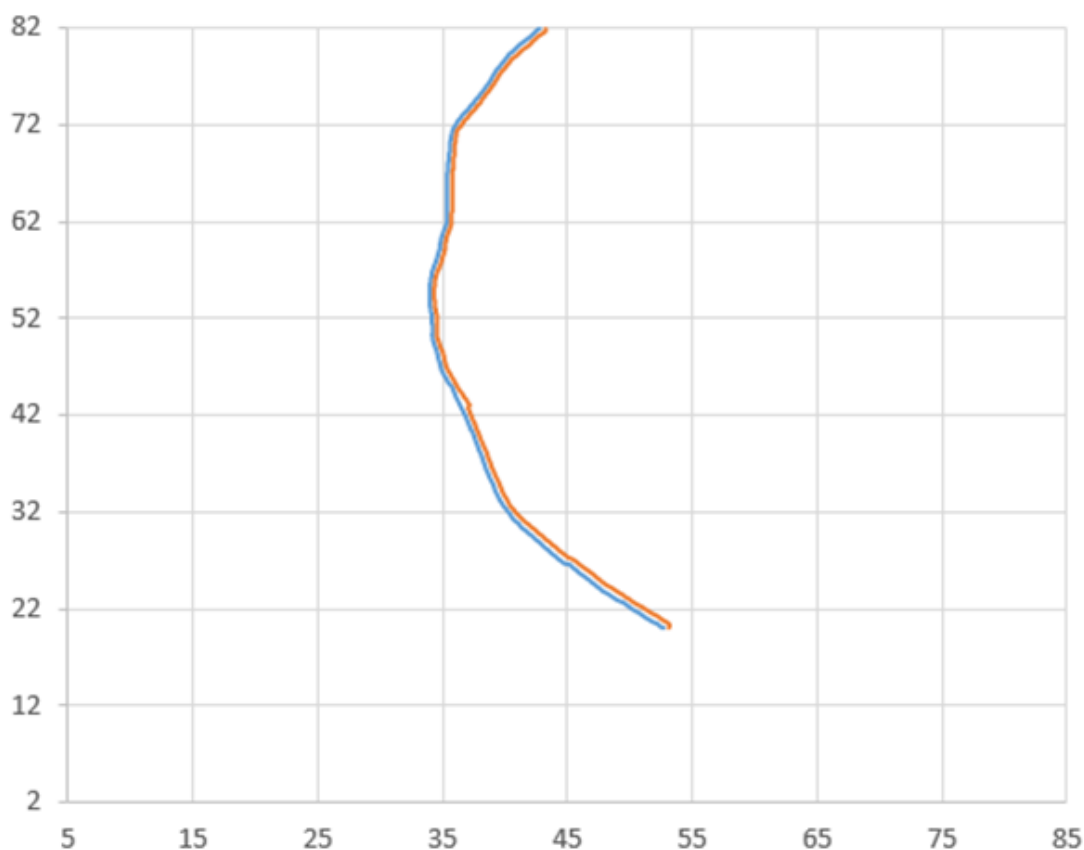


Рис. 6. График координат границ трещины

Для повышения автономность робота, передача координат  $(U_x; U_y)$ , из файла MS Excel будет производиться по wi-fi, в качестве модуля приёмника выбран: ESP8266.

### *Заключение*

Используя глубокую сверточную нейронную сеть (CNN) с архитектурой UNet, мы обучили её на основе базы данных нескольких сотен изображений фотографий, содержащих различные дефекты в элементах металлических конструкций.

На основе языка Python вывели координаты границ трещины и конвертировали их в значения напряжения, необходимые для работы робота.

На основе языка Arduino IDE составили программу для работы робота.

Учитывая актуальность работы и доступность модулей Arduino данный концепт является оптимальным решением поставленной проблемы.

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. – 2014.
2. Ronneberger O., Fischer F., Brox T. U-Net: Convolutional networks for biomedical image segmentation // International Conference on Medical Image Computing and Computer-Assisted Intervention. – 2015. Vol. 9351. pp. 1097–1105.

3. Unet 3+: A full-scale connected Unet for medical image segmentation / H. Huang, L. Lin, H. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, J. Wu – 2020.
4. DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation / R. Li, W. Liu, L. Yang, S. Sun, W. Hu, F. Zhang, W. Li – 2017.
5. Монк, Саймон. Програмируем Arduino: профессиональная работа со скетчами / Саймон Монк ; [перевел с английского А. Киселев]. – Санкт-Петербург: Питер, 2017. – 272 с.
6. Шишкин, Г. Г. Электроника : учебник для бакалавров / Г. Г. Шишкин, А. Г. Шишкин. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2019. – 703 с.

© Д. В. Третьяков, Р. И. Чижменко, В. В. Паксютов, 2022